THESIS / THÈSE

MASTER EN SCIENCES INFORMATIQUES

Contribution à l'élaboration d'une sémantique formelle pourCONGOO

Géron, Jean-Marc

Award date:
1996

Awarding institution:
Universite de Namur

FACULTES
UNIVERSITAIRES
NOTRE-DAME DE LA PAIX
NAMUR

INSTITUT D'INFORMATIQUE

Contribution à l'élaboration d'une sémantique formelle pour CONGOO

Jean-Marc Géron

Promoteur :
Professeur E. Dubois

Mémoire présenté par
Jean-Marc Géron
en vue de l'obtention du grade
de Licencié et Maître en Informatique

Année académique 1995-1996

RESUME

Les modèles conceptuels de données courants tels que le modèle Entité-Association ne permettent pas de structurer efficacement les données géographiques. Dans ce contexte, un nouveau modèle conceptuel de données a été développé par Pantazis. Le thème de ce mémoire est l'étude de ce nouveau modèle conceptuel de données géographiques appelé CONGOO. Ce modèle de données propose des solutions intéressantes pour les problèmes de modélisation conceptuelle des données géographiques mais il ne possède pas de sémantique précise. Notre principal objectif est de donner une sémantique formelle à ce modèle de données. Pour définir formellement cette sémantique, nous utilisons un langage formel de spécification appelé GLIDER.

ABSTRACT

Current conceptual data models, like the Entity-Relationship model, do not make possible to effiently structure the geographical data. In this context, a new conceptual model of data has been developed by Pantazis. The topic of this thesis is the study of this new conceptual model of geographical data. This model of data proposes some interesting solutions to the conceptual modeling problems of geographical data but it hasn't got a precise semantics. Our main target is to give a formal semantics to this model of data. In order to define this semantics formally we have used a formal language specification called GLIDER.

Remerciements

Nous tenons à exprimer toute notre gratitude à Monsieur E. Dubois qui nous a assisté et judicieusement conseillé pour mener à bien cette étude.

Nous remercions également Monsieur D. Pantazis pour sa grande disponibilité et pour nous avoir fait part des précieux résultats de ses recherches.

Nous remercions encore Monsieur J.P. Donnay pour son accueil et le prêt de nombreux ouvrages.

Enfin, nos remerciements s'adressent encore à ceux qui nous sont proches et qui n'ont jamais cessé de suivre l'évolution de notre travail et de nous accorder leur soutien. Nous pensons en particulier à nos parents qui nous ont prodigué aide et encouragement tout au long de ce travail.

Table des matières

Introduction

La majorité des récentes méthodes de développement de systèmes d'information informatisés distingue deux activités de base : l'ingénierie des besoins (aussi appelé analyse des besoins) et l'ingénierie logiciel (parfois appelée activité de conception) .

Lors de la première activité, le concepteur du système d'information doit s'attacher à cerner et à comprendre au mieux les besoins des futurs utilisateurs. La seconde activité consiste à transformer ces besoins en code exécutable.

Dans le cadre de notre travail, nous nous intéressons plus particulièrement à la première activité et plus précisément à la modélisation de l'information géographique dans le cadre du développement d'un système d'information géographique.

Les tâches de l'ingénierie des besoins

L'activité d'ingénierie des besoins est loin d'être une simple retranscription des désirs des futurs utilisateurs. En effet, le but de l'analyste est de spécifier les besoins pour produire des documents qui serviront comme énoncé d'un problème à résoudre, comme contrat entre client et développeur, comme communication entre concepteurs, clients et utilisateurs et comme validation de la conception. Il convient donc de mettre en place un processus interactif et répétitif entre le client et l'analyste afin de progressivement clarifier les désirs du client qui souffrent en général d'imprécision, d'incohérence et d'un manque de structuration.

Cette activité peut être divisée en quatre tâches pour l'analyste [Dubois, Du Bois et Petit 93, p.461] (Figure 1).

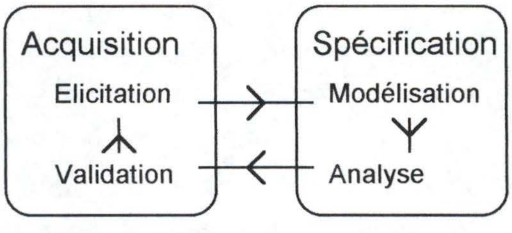


Figure 1 : Les tâches de l'activité d'ingénierie des besoins

1. Durant la tâche d'élicitation, l'analyste collecte l'information sur les désirs et les problèmes du client. Cette collecte peut prendre différentes formes : Interviews, discussions, études de divers documents.

2. Pendant la tâche de modélisation, l'analyste traite les informations collectées et élabore un modèle.

3. Durant la tâche d'analyse, l'analyste met en évidence des problèmes dans le modèle tels que contradiction, ambiguïté, redondance, non-minimalité (sur spécification), incomplétude (sous spécification).

4. Lors de la tâche de validation, l'analyste vérifie l'adéquation de sa modélisation avec le client et discute des problèmes découverts pendant la tâche d'analyse.

Importance de l'activité d'ingénierie des besoins

L'activité d'ingénierie des besoins est très importante dans le développement des systèmes d'information car les éléments mis en évidence lors de cette activité constituent les bases de toute la conception d'un système d'information. Une erreur introduite à ce stade du développement peut être découverte seulement quand le logiciel est terminé. Ces erreurs de développement sont donc en général les plus coûteuses à corriger car elles peuvent demander une reconception et une réimplantation d'une large part du système d'information. Ces types d'erreurs sont en général les plus nombreuses. Il est donc important de mettre tout en oeuvre pour réussir au mieux cette activité.

Les langages de spécification des besoins

La plupart du temps et surtout il y a quelques années, l'analyste utilisait le langage naturel pour spécifier les besoins. L'utilisation du langage naturel a l'avantage d'être communicable à toutes les personnes intéressées par le développement du système d'information mais il a le défaut majeur qu'il peut être compris et interprété de manières différentes selon les lecteurs.

Les analystes se sont donc souvent tournés vers des méthodologies (MERISE, IDA, REMORA, ...) fondées sur des modèles rigoureux tels que E-A, DFD, Analyse OO. Ces langages possèdent une syntaxe formelle et une interprétation formelle incomplète.

Dernièrement, les recherches dans ce domaine ont identifié les besoins d'un langage de spécification des besoins et ils ont notamment montré que les langages rigoureux n'ayant pas une sémantique formelle complète laissent des ambiguïtés et ne permettent pas de concevoir des outils performants. C'est pourquoi d'autres langages ont été développés tels que RML [Greenspan, Bordiga et Mylopoulos 86], ERAE [Dubois, Hagelstein et Rifaut 90], GLIDER [Dubois, Du Bois, Rifaut et Wodon 91]. Ces langages disposent de règles d'interprétation formelle qui permettent de garantir l'absence d'ambiguïté, de redondance, de contradiction. De plus, ils permettent souvent l'utilisation d'outils performants pour supporter l'activité d'ingénierie des besoins.

Objectifs du mémoire

Dans le cadre de la création d'une méthode de conception (analyse/conception), appelée MECOSIG, pour faciliter le développement des systèmes d'information géographique, un nouveau modèle de structuration de l'information géographique à été élaboré (appelé CONGOO).

L'objectif premier de notre travail est de relever les caractéristiques que devrait proposer un bon modèle de structuration de l'information géographique pour être utilisé dans l'activité d'ingénierie des besoins et d'observer si le modèle CONGOO les possède.

L'objectif second est de donner une sémantique formelle à ce modèle afin de relever si, dans les définitions des concepts du modèle, il y a des ambiguïtés, des contradictions, des redondances ou des silences.

Structure du mémoire

Dans le premier chapitre, nous définissons ce qu'on entend par systèmes d'information géographique et identifions leurs caractéristiques spécifiques par rapport aux systèmes d'information "traditionnels".

Ensuite dans le deuxième chapitre, nous étudions si la spécificité des systèmes d'information géographique réclame des méthodes de conception et des modèles de structuration de l'information propre. Nous terminons ce chapitre en présentant et en critiquant le modèle CONGOO.

Dans le troisième chapitre, nous donnons une sémantique formelle au modèle CONGOO en définissant ses concepts à l'aide d'un langage de spécification formel.(GLIDER).

Enfin dans le dernier chapitre, nous étudions en détail les relations topologiques[1] qui jouent un rôle majeur dans la modélisation de l'information géographique.

---

[1] Les relations topologiques "[...] are those which are invariant under topological transformations, i.e., they are preserved if the objects are translated, rotated, or scaled." [Egenhofer 91]

Chapitre 1
Les systèmes d'information géographique

1.1 Introduction

Au cours des deux dernières décennies, l'utilisation des systèmes d'information géographique dans les organisations n'a cessé de croître. Par exemple, en France, Laurini parle d'un taux d'expansion du chiffre d'affaires du marché du domaine des systèmes d'information géographique de l'ordre de 30 à 50 % [Laurini et Milleret-Raffort 93]. Au Canada, en 1990, 1355 firmes privées sont dénombrées touchant à ce domaine, soit une augmentation de 18% depuis 1985. Au cours des derniers exercices, le secteur privé a offert des services et de l'équipement logiciel pour 750 millions de dollars canadiens annuellement, soit deux fois plus que pour la période de 1979-1983 [Jaton et Bédard 93]. Certains analystes prévoient que les systèmes d'information géographique constitueront l'un des moteurs du développement de l'informatique au cours des années 90 : d'après Daratech Inc. le marché mondial des systèmes d'information géographique devrait atteindre 10,6 milliards de dollars US en 1994 et à peu près en même temps Smallword estime le marché européen, à peine naissant, à 400 millions de dollars US, ce qui ouvre des perspectives de croissance considérable [Rouet 91].

Plusieurs raisons expliquent ce succès croissant.

Tout d'abord, l'explosion fantastique de la production de l'information géographique due principalement aux progrès des technologies d'acquisition telles que les levés topographiques, l'imagerie satellitaire, la photogrammétrie aérienne, les systèmes de positionnement (GPS), la digitalisation, le scannage de cartes et de plans,... a permis un plus grand nombre d'applications et demandé des systèmes plus sophistiqués de gestion de cette information.

Ensuite, l'évolution considérable des techniques d'analyse et de représentation graphique a augmenté les possibilités qu'offrent les systèmes d'information géographique.

De plus, la rapide réduction des prix du matériel et logiciel informatique de ces dernières années a contribué à l'expansion continue des systèmes d'information géographique.

Enfin, les systèmes d'information géographique offrent de nombreux avantages aux utilisateurs de données spatialement référencées par comparaison avec l'utilisation des cartes papiers traditionnelles. Ils permettent non seulement d'assurer le dessin des objets géographiques mais également de les traiter par eux-mêmes en tant que tels, en les identifiant par un code, en les décrivant à l'aide d'informations alphanumériques, en les positionnant dans l'espace, en décrivant leurs formes géométriques, et en prenant en compte les relations des objets entre eux.

Dans ce chapitre, nous essayons de définir ce qu'on entend par systèmes d'information géographique et de souligner leurs spécificités par rapport aux systèmes d'information "traditionnels".

1.2. Définition[1]

Il existe de nombreuses définitions du terme système d'information géographique (S.I.G.) dans la littérature.

" A geographic information system is any manual or computer based set of procedures used to store and manipulate geographically referenced data "[Aronoff 89, p.39].

" A GIS is a computer system - hardware and software - that helps analysts discover relationships between and among sets of computer - readable, geographically referenced data that they could not see or understand easily without the aid of this technology" [Marx 91, p.413].

"Un systéme d'information géographique (S.I.G.) concerne les techniques et méthodes d'acquisition d'informations référencées spatialement, leur codage, leur organisation en banque de données, leur traitement dans le but de fournir notamment aux ingénieurs et aux planificateurs des informations complexes pour le dimensionnement d'ouvrages et pour l'aide à la décision" [Calloz 91, p.54] cité dans [Pantazis 94, p.146].

"Le système d'information géographique est un système de gestion de base de données pour la saisie, le stockage, l'extraction, l'interrogation, l'analyse et l'affichage de données localisées" [Laurini et Milleret-Raffort 93, p.51].

"Un S.I.G. (Système d'information géographique) est identifié comme étant un logiciel offrant des capacités de cartographie numérique, de gestion de base de données et d'analyse spatiale" [Archamboult et Bédard 90] cité dans [Pantazis 94, p.148].

---

[1] Cette partie est inspirée en partie du chapitre 4 de la thèse de doctorat de Pantazis [Pantazis 94].

"A digital information system whose records are somehow geo graphically referenced" [Goodchild 91, p.117].

Plusieurs auteurs ont montré que les différentes définitions trouvées dans la littérature sont souvent contradictoires et ambiguës. Les quelques définitions reprises ci-dessus soulignent à nouveau ces différences. Par exemple à ce propos, Donnay écrit "..une définition non équivoque et communément admise des S.I.G. [...] n'existe pas. La littérature propose plusieurs, on devrait dire quantité de définitions plus ou moins distinctes. Aucune n'est fausse, mais aucune n'est complète. Cela vient vraisemblablement du caractère multidisciplinaire des S.I.G., prévalant tant à ses origines que dans ses domaines d'application" [Donnay 92, p.1].

Toutefois comme le relève Pantazis dans sa thèse de doctorat "..on peut constater que, à l'unanimité, tous les auteurs s'accordent sur le fait que les S.I.G. concernent (entre-autres) les données géographiques ou spatiales, ou à références spatiales, ou géo-référencées, ou localisées ou géo-codées etc.. et les traitements sur ces données " [Pantazis 94, p.150].

Les données géo-graphiques

Les données spécifiques que gèrent les systèmes d'information géographique sont des données ayant une représentation "graphique" référencée spatialement. Ces données sont appelées par Pantazis "données géo-graphiques". Soulignons que ces données graphiques référencées spatialement ne font pas nécessairement référence à de l'information "géographique" à proprement parler.

Bien sûr les systèmes d'information géographique gèrent aussi en général des données alphanumériques géo-référencées ou non. Elles constituent souvent des données attributaires d'objets ou phénomènes référencés spatialement.

L'information associée à un objet ou phénomène référencé spatialement est donc composé en général de deux types de données :

- Les données qui servent à localiser et à décrire la forme de l'objet ou du phénomène c'est-à-dire les données géo-graphiques.
- Les données descriptives appelées aussi données thématiques qui servent à décrire l'objet ou le phénomène.

Les données géo-graphiques ont des caractéristiques particulières différentes des données alphanumériques. Ci-dessous nous en relevons quelques-unes [Laurini et Milleret-Raffort 93].

Tout d'abord, pour les données géo-graphiques à caractère cartographique il y a un lien très fort entre la sémantique et l'échelle des données car, en cartographie, seuls les objets que l'ont peut distinguer sont représentés. Par exemple, sur une carte de France tenant dans une page A4, on ne représentera pas les 36000 communes de France. Il est donc notamment indispensable de connaître l'échelle d'origine d'une carte vu qu'il est possible informatiquement de tracer des cartes à n'importe quelle échelle.

Ensuite, en général toute donnée géo-graphique a une certaine qualité liée à la précision variable des instruments de mesures et à des méthodes dites de généralisation. Notons que l'échelle et la précision sont généralement associées, car le choix de l'échelle de restitution dicte souvent la précision à rechercher au niveau de la saisie des données. Il faut donc faire attention que l'information soit de même qualité pour ne pas avoir des résultats erronés.

La représentation géométrique d'un objet géographique peut varier selon l'utilisateur (Figure 1.1).

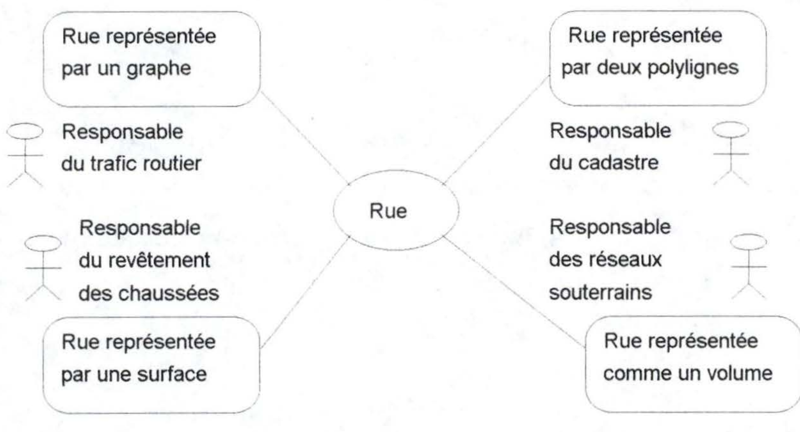


Figure 1.1 Différentes visions d'une rue [Laurini et Milleret - Raffort 93,p.44]

Pour assurer une bonne lisibilité et une bonne compréhension d'ensembles d'objets ou phénomènes spatiaux, il faut choisir convenablement les symboles, les couleurs, le placement des textes...

Pour représenter les objets et les phénomènes spatiaux, il est nécessaire de s'appuyer sur un certain nombre de concepts issus de la géométrie euclidienne, de la théorie des graphes et de la topologie.

Pour représenter mathématiquement et informatiquement des objets et des phénomènes géo-référencés spatialement, nous devons leur donner une définition en intension[1] car il est impossible de leur donner une définition en extension vu qu'ils sont composés en général d'un ensemble infini de points. Par exemple un segment de droite est composé d'un ensemble infini de points. Ce qui posera parfois des problèmes pour savoir si un objet appartient à l'ensemble.

La représentation graphique des objets spatiaux se fait sous deux formes principales : la première est sous forme de vecteur et la deuxième est sous forme raster appelée aussi matriciel (Figure1.2.). La première décrit les objets spatiaux par leurs délimitations sur le territoire à l'aide d'entités géométriques élémentaires telles que point, ligne, surface... Elle est en général obtenue par la digitalisation de carte. Par contre la deuxième représente les objets d'un territoire en associant une ou un ensemble de valeurs à chaque surface élémentaire d'une maille régulière couvrant ce territoire. Cette forme de représentation graphique peut être issue par exemple du scannage d'une carte.

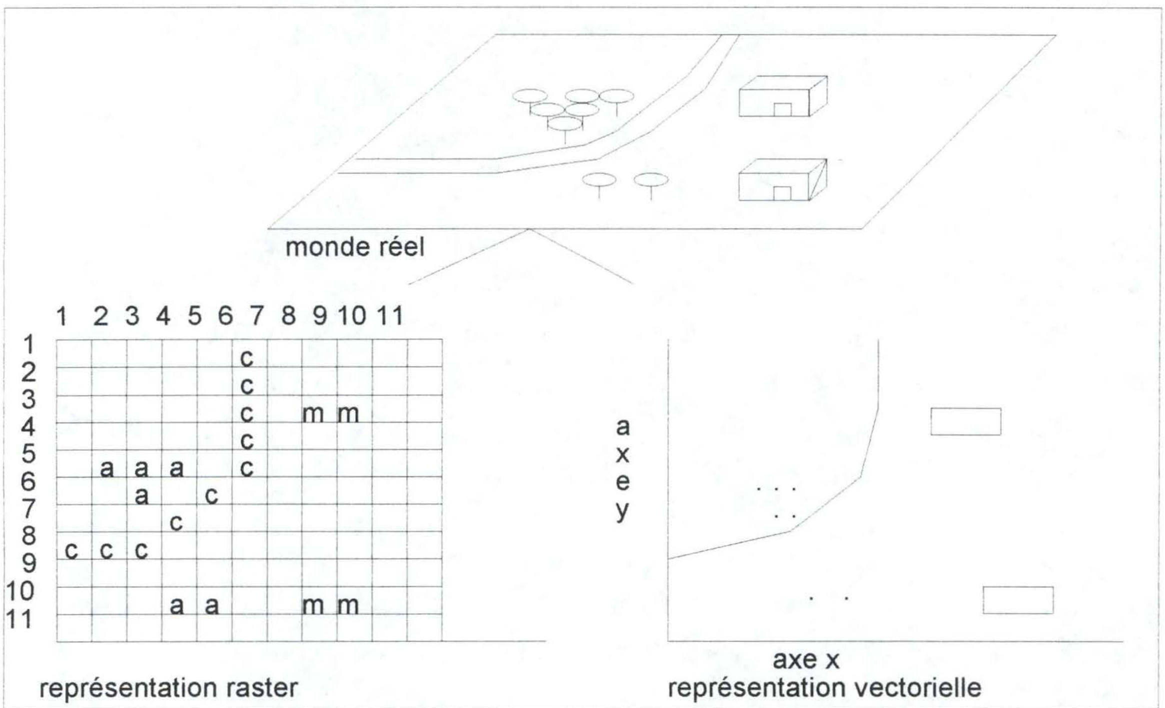


Figure 1.2 : Comparaison de la représentation sous forme vecteur et raster

[1] Définir un ensemble en intension signifie que l'on définit l'ensemble par ces propriétés par contre définir un ensemble en extension signifie que l'on donne la liste de tous les éléments constituant cet ensemble.

Un système d'information particulier

De même que pour les systèmes d'information géographique, on peut trouver dans la littérature de nombreuses définitions pour le terme système d'information (S.I.). Dans ce mémoire nous ne discuterons pas ces définitions car cela nous éloignerait de notre sujet. Nous nous contenterons de reprendre les propos de Bodart et Pigneur relativement bien admis [Bodart et Pigneur 93, p.1].

> "Par système d'information d'une organisation, nous entendons une construction formée d'ensembles :
>
> - d'informations, qui sont des représentations - partielles - de faits qui intéressent l'organisation;
> - de traitements, qui constituent des procédés d'acquisition, de mémorisation, de transformation, de recherche, de présentation et de communication des informations;
> - de règles d'organisation qui régissent l'exécution des traitements informationnels;
> - de ressources humaines et techniques requises pour le fonctionnement du S.I."

Dans la littérature, le terme S.I.G. est souvent employé pour nommer le progiciel ou logiciel spécialisé qui gère les données géo-graphiques et alphanumériques (Par exemple : Arc/Info, Sicad, ...). Dans cette approche, il joue le rôle de boîte à outils informatiques à l'intérieur de l'organisation. D'après la définition de S.I. vu ci-dessus, il est clair que cette approche du S.I.G. ne peut être considérée comme un véritable S.I. car il ne considère que le logiciel qui peut manipuler l'information géographique. Cette approche du S.I.G. est en fait fort réductrice et n'est qu'une partie d'un S.I.G. informatisé.

Une autre approche le considère comme un ensemble d'informations, de traitements, de ressources humaines, matérielles, et financières intégré au sein d'une organisation pour répondre au besoin d'informations de celle-ci. Il dépasse alors largement la notion d'un simple logiciel. Dans ce cas le S.I.G. peut être considéré comme un S.I. particulier. Le S.I.G. n'est donc pas réduit dans cette deuxième approche au seul logiciel. Notons qu'un S.I.G. peut être exclusivement manuel.

Pour ne pas confondre ces deux approches dans la suite de notre travail nous appellerons S.I.G.-logiciel quand il s'agira du S.I.G. vu comme dans la première approche. Une définition du S.I.G.-logiciel est donnée par Pantazis.

"S.I.G.-logiciel = Logiciel (plutôt progiciel) spécial, combinant le graphisme et la gestion des bases de données alphanumériques et géo-graphiques, offrant des fonctionnalités d'analyse spatiale" [Pantazis 94, p3].

sont utilisées[1] pour l'informatisation d'un S.I.G. sont principalement celles d'analyse spatiale et celles offertes par les systèmes de gestion de bases de données (SGBD) géo-graphiques et alphanumériques.

1.3.1. Le sous-système d'acquisition des informations

Ce sous-système convertit l'information spatiale en une forme qui peut être stockée par le S.I.G.-logiciel. La conversion de format, la détection et la correction des erreurs, la reconstruction de la topologie ou la généralisation font aussi parfois partie de ce processus.

Trois situations peuvent se présenter pour les données que l'on désire introduire dans les S.I.G.

1. Il n'existe aucun document, plan ou carte représentant les objets ou les phénomènes à traiter même sur support papier traditionnel. Dans ce cas ils peuvent être acquis directement par imagerie satellitaire, photo aérienne, carnets électroniques de terrains.

2. Si les données à traiter sont disponibles sur supports papiers tels que cartes, plans, graphiques alors le problème à résoudre est de numériser ces données. Deux moyens principaux existent pour numériser ces documents : saisie par digitalisation et saisie par scanner.

3. Les données à traiter sont déjà numérisées. Pour qu'elles soient utilisables il faut que leur format soit connu par le S.I.G. . Si il ne l'est pas, on utilise des interfaces qui permettent

- soit de transformer directement les données dans le format interne du S.I.G. récepteur grâce à des bibliothèques de conversion à ce format interne.

- soit de passer par l'intermédiaire d'un format d'échange reconnu par une fonction d'importation de données du S.I.G. récepteur.

En ce qui concerne les données alphanumériques elles sont comme pour les S.I. "traditionnels" saisies par l'intermédiaire d'un clavier ou importées par exemple sous forme de fichier ASCII.

Une des difficultés de ce processus est de bien intégrer ces données hétérogènes, de qualité, de fiabilité, de précision et d'extension spatiale bien différentes.

---

[1] Les S.I.G.-logiciel sont des boites d'outils informatiques puissants dont l'installation permet d'informatiser en partie le S.I.G. mais aussi souvent une partie du système opérant de l'organisation.[Pantazis 94]

La définition de S.I.G. que nous admettons pour notre travail est

Un système d'information géographique (S.I.G.) est un Système d'information (S.I.) qui traite des données ayant une représentation graphique référencée spatialement.

Les domaines d'application des S.I.G. sont très variés : urbanisme, environnement, transport, archéologie, tourisme, géologie, géotechnique, géomarketing, applications militaires, cartographie, gestion des réseaux souterrains, planification des ressources,...

Signalons que beaucoup d'autres termes sont trouvés dans la littérature pour nommer des systèmes d'information géographique qui se différencient principalement par le but de leur création. Par exemple nous trouvons : système d'information du territoire, système d'information à référence spatiale, système d'information urbain, système d'information des ressources naturelles, ...

1.3. Problèmes et solutions informatiques particulières des systèmes d'information géographique informatisés

Dans cette partie nous présentons les différentes caractéristiques des S.I.G.-informatisés en insistant sur les différences avec les S.I."traditionnels". Par souci de clarté, nous avons divisé le S.I.G. en sous-systèmes.

Nous avons distingué quatre sous-systèmes avec des fonctionnalités différentes .

1. Sous-système d'acquisition des informations
2. Sous-système de gestion de l'information
3. Sous-système de transformation et d'analyse de l'information
4. Sous-système de présentation et diffusion de l'information

Avant de présenter chaque sous-système, il faut savoir qu'une des différences fondamentales avec les S.I.traditionnels est que le principal moyen d'informatisation d'un S.I.G. est l'intégration d'un S.I.G.-logiciel dans l'organisation. Les fonctions d'un S.I.G.-logiciel qui

Nous constatons donc que le processus d'acquisition des données géo-graphiques est différent de l'acquisition alphanumérique.

Tout d'abord, il demande souvent des équipements variés et plus nombreux que pour l'acquisition de données alphanumériques.

Ensuite, les modes d'acquisition sont très différents. Nous avons vu que certaines données peuvent être directement mesurées sur le terrain ou captées à distance ou saisies à partir de cartes ou de plans existants, ou récoltées par des organismes de production de données et ensuite importées.

Enfin, il faut savoir que le processus d'acquisition des données est souvent lent. En effet il peut prendre d'un mois à plusieurs années pour avoir toutes les données complètes dans le bon format.

Nous pouvons dire pour conclure que deux des caractéristiques des données géo-graphiques est d'être multimédia et multisource vu la diversité de leurs modes d'acquisition.

1.3.2. Le sous-système de gestion de l'information

Ce sous-système inclut les fonctionnalités classiques d'un système de gestion de base de données et est réalisé par le "coeur" du S.I.G.-logiciel.

Nous avons vu que les objets et phénomènes spatiaux étaient toujours représentés par des données géo-graphiques et souvent avec des données thématiques associées. Ces deux types de données doivent être gérés efficacement par le S.I.G.-logiciel. Pour stocker les données thématiques, il n'y a pas de problème, un SGBD standard (par exemple relationnel) peut être utilisé.

Par contre pour les données géo-graphiques, l'utilisation d'une base de données relationnelles est beaucoup plus difficile. Il est vrai que les coordonnées en x et y des contours des objets, y compris les liens plus ou moins complexes entre objets directement issus de leur géométrie peuvent être introduits dans les tables d'une base de données relationnelles mais les fonctions dont on a besoin pour manipuler les données géo-graphiques ne sont pas bien assurées par les SGBD relationnel standard.

Plusieurs problèmes peuvent être mis en évidence :

Un premier problème est que la base de données reçoit et retourne les valeurs de x et y exactement comme d'autres valeurs numériques; or ce n'est pas du tout ce que l'on veut. Ce que l'on souhaite c'est de voir l'effet de ces coordonnées x et y c'est-à-dire de voir se dessiner sur l'écran le plan ou la carte des objets sélectionnés. Or la BD ne sait pas le faire directement. On pourrait se dire qu'il suffit de faire un programme en mode graphique avec les données renvoyées par la BD. Mais les SGBD relationnels seraient incapables d'envoyer assez vite toute la masse d'informations pour remplir directement un écran graphique à haute résolution [Rouet 91].

Un deuxième problème majeur de l'usage des SGBD classiques est que le langage d'interrogation de la base de données relationnel ne permet pas ou avec beaucoup de difficultés de traiter des requêtes comportant des critères géométriques ou spatiaux tels que la proximité, l'appartenance, l'adjacence,... Ainsi par exemple, le langage classique d'interrogation SQL ne permet pas ou très difficilement de rédiger des requêtes spatiales qui font appel au positionnement et aux opérateurs géométriques car ce langage ne prend pas en compte les aspects intensionnels des données géo-graphiques [Laurini et Milleret-Raffort 93], [Aronoff 89].

Enfin un autre problème est la gestion de la haute interrelation entre les données des S.I.G. qui requiert un système de sécurité beaucoup plus sophistiqué que l'approche verrouillage des enregistrements utilisée par les SGBD "traditionnels" [Laurini et Milleret-Raffort 93], [Aronoff 89]. Pour assurer l'intégrité de la base de données géographiques, le système de sécurité doit protéger l'intégrité de multiples fichiers dans lesquels des données spatiales sont stockées. Un changement dans un enregistrement peut créer de multiples erreurs. Deux types de lien existent entre les données.

Tout d'abord il y a un lien entre les données géo-graphiques et leurs données attributaires. Il faut donc pouvoir répercuter sur les données géo-graphiques des actions ou des analyses menées sur les données attributaires et, inversement, il faut répercuter en mode attributaire des données déduites de la géométrie modifiées par la mise à jour ou enrichies par l'analyse spatiale.

Ensuite il y a les liens entre les données géo-graphiques. Il s'agit de liens entre objets graphiques (souvent de nature topologique) qui peuvent être déduits de la géométrie ou imposés par l'utilisateur et qui ne sont pas forcément gérés directement au niveau graphique.

Les solutions

La difficulté à utiliser les SGBD existant pour manipuler les données géo-graphiques a requis que des SGBD hybrides ou modifiés soient développés [Aronoff 89], [Larue, Pastre, viémont 93].

Actuellement, les S.I.G.-logiciels sont basés sur une des quatre architectures suivantes (Figure 1.3).

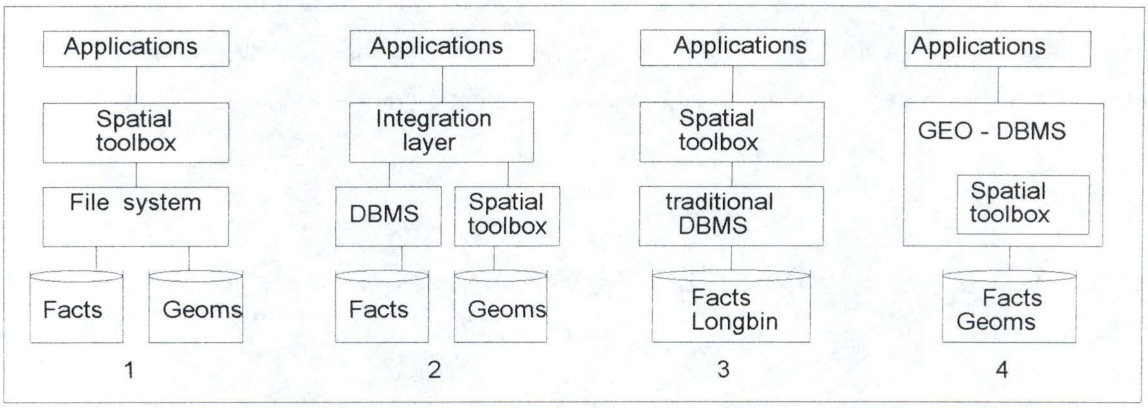


Figure 1.3 : Les quatre principales architectures des S.I.G.-logiciels [Larue, Pastre et Viemont 93]

La première architecture est basée sur des systèmes construits directement sur un ensemble de fichiers. Les fonctionnalités réalisées par ce système sont très limitées avec peu d'évolution de capacité.

La deuxième architecture utilise un SGBD traditionnel, généralement relationnel, pour les données non géo-graphiques et un système à part pour stocker et manipuler les données géo-graphiques. Le S.I.G.-logiciel ARC-INFO, le premier du marché mondial appartient à cette catégorie. Un objet spatial, caractérisé par des données géo-graphiques et des données thématiques, a une partie de son information dans chacun de ces deux systèmes. Vu que les deux systèmes de stockage possèdent leur propre langage d'interrogation et mécanisme d'intégrité de données, une couche intégration est nécessaire dans l'architecture pour par exemple pouvoir traiter une requête liant des propriétés spatiales et non spatiales d'objets. L'inconvénient majeur de ce système est que même si les performances de chaque système sont satisfaisantes celles-ci sont affectées par la couche d'intégration.

La troisième architecture gère les données spatiales et non spatiales dans une même base de données. Pour stocker l'information spatiale dans un modèle relationnel, il est nécessaire de scinder les entités géographiques en leurs différentes composantes qui seront stockées dans des tables séparées. La recherche des objets complets est alors réalisée par l'intermédiaire de jointure de relations, ce qui rend le système très lent et d'utilisation complexe (voir ci-dessus).

De plus il est impossible d'assurer l'intégrité des données spatiales. L'utilisateur est libéré du problème posé par la formulation complexe des requêtes par une couche supérieure au SGBD conventionnel. Cette couche traduit les requêtes géographiques en des requêtes comprises par le SGBD.

La quatrième architecture, récemment apparue, est basée sur un SGBD extensible permettant d'ajouter des types spatiaux ainsi que les fonctions et prédicats spatiaux associés. Elle enlève les aspects de dysfonctionnement et de dualité des langages des architectures précédentes. Le grand avantage de ce système est leur performance. En effet, l'implémentation directe de nouveaux types de données dans le noyau du SGBD les rend très efficaces.

1.3.3. Le sous-système de transformation et d'analyse de l'information

Ce système est assuré par le S.I.G.-logiciel. En général les S.I.G.-logiciels peuvent supporter différents types de traitements statistiques et différentes analyses spatiales. Les traitements statistiques les plus fréquents sont : tri sur une variable, statistique sur une variable (tel que calcul de moyenne, écart-type,..), génération de graphiques, définition de bornes de classes. Les outils d'analyses spatiales des différents S.I.G. logiciels existants sont rares et très récents. Les méthodes les plus fréquentes sont l'interpolation, l'extrapolation, la théorie des graphes, la topologie, les méthodes issues de la recherche opérationnelle, celles issues des techniques de recouvrement de cartes et des méthodes issues de la simulation et de l'optimisation multicritère.

1.3.4. Le sous-système de présentation et de diffusion de l'information

Ce sous-système est aussi assuré en partie par le S.I.G.-logiciel. De nombreux formats de présentation sont en général disponibles : les représentations cartographiques simples (2D), les blocs-diagrammes et cartes (3D), les vues en coupe, les vues en perspective, les histogrammes et les graphiques statistiques. En général les S.I.G.-logiciels proposent des sorties numériques et textuelles, ainsi que des sorties cartographiques. Cette tâche peut être réalisée par des matériels divers : écran, table traçante, scanner d'écriture ou imprimante graphique.

1.4. Conclusion

Nous avons vu que derrière le terme S.I.G. se cachent différentes significations. Une signification le considère comme un véritable S.I. d'organisation qui traite des données particulières qui sont les données géo-graphiques. Ces données géo-graphiques ont des caractéristiques différentes des données alphanumériques et des modes d'acquisition très variés. En ce qui concerne l'informatisation des S.I.G. il faut retenir qu'elle se fait principalement par l'intégration d'un S.I.G.-logiciel dans l'organisation. Celui-ci ne peut pas être basé seulement sur un système de gestion de base de données "traditionnel" pour offrir une gestion efficace des données. En d'autres termes nous avons montré que bien qu'un S.I.G. soit un véritable S.I., il demande des solutions particulières pour son informatisation vu les caractéristiques des données qu'il doit traiter.

Signalons pour terminer que la recherche des solutions particulières que demandent les traitements des données spatiales a donné naissance à une nouvelle discipline appelée géomatique. L'office de la langue française du Québec définit ce terme comme suit : "Discipline ayant pour objet la gestion de données à référence spatiale par l'intégration des sciences et des technologies reliées à leur acquisition, leur stockage, leur traitement et leur diffusion." [Jaton et Bédard 93]

Chapitre 2
Les modèles conceptuels pour structurer l'information géographique

2.1. Introduction

Dans ce chapitre nous examinons si les méthodes de conception de S.I. traditionnel sont adaptées à la conception des S.I.G.. Ensuite nous étudions plus en détails les besoins des modèles conceptuels de structuration de l'information pour être appliqué à l'information géographique. Pour terminer nous présentons et critiquons le formalisme de structuration de l'information géographique : CONGOO.

2.2. Les méthodes de conception des S.I.G.

Avant de discuter brièvement des méthodes de conception des systèmes d'information géographique, rappelons le but d'une méthode de conception de S.I. et les composants indispensables de celle-ci.

Depuis une vingtaine d'années un effort considérable a été fait dans la mise au point de méthodes de développement des S.I. dans le but de promouvoir une véritable approche industrielle du développement des systèmes d'information. Ces méthodes permettent de sortir de l'empirisme et évitent un grand nombre de problèmes tels que les dépassements presque systématiques des plannings et des budgets, coûts de maintenance très élevés, redondance de tâches de conception,...

Pour présenter les composants indispensables d'une méthode de conception de S.I. reprenons les propos de Bodart et Pigneur.

"Il n'est plus discutable actuellement que toute méthode doit proposer une démarche fondée sur des modèles et mise en oeuvre à l'aide d'outils logiciels (Figure 2.1).
[...] Un modèle est formé de concepts et de règles relatives à leur utilisation. [Bodart et Pigneur 93, p.8]

"La démarche proposée par une méthode de conception est constituée, essentiellement, d'un ensemble de règles qui assurent la guidance et le contrôle dans l'application des modèles proposés pour définir et spécifier un S.I. ainsi que dans l'usage des outils logiciels qui supportent la mise en oeuvre de ces modèles."[Bodart et Pigneur 93, p.8]

Soulignons que la démarche proposée doit être un cadre adaptable au contexte particulier d'une organisation et d'un projet.

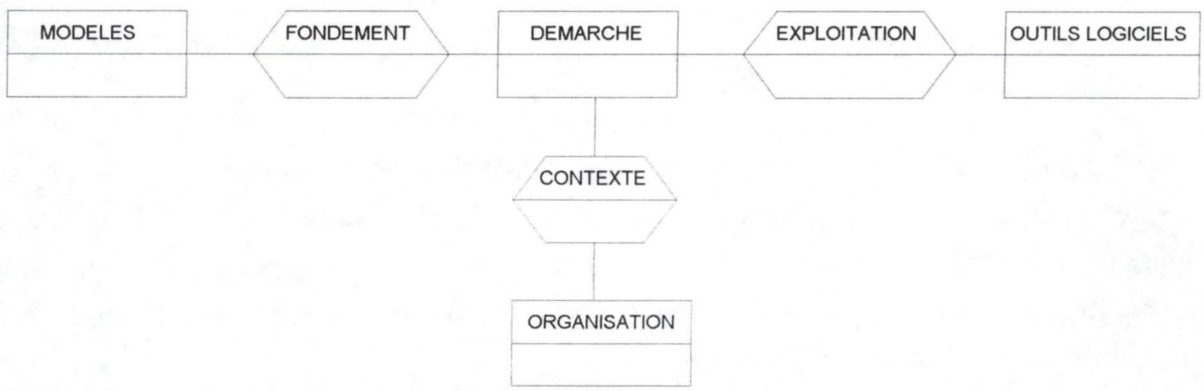


Figure 2.1 Les pôles d'une méthode de conception d'un S.I. Une démarche fondée sur des modèles et exploitée à l'aide d'outils logiciels dans le contexte d'une organisation [Bodart 93].

Pour les S.I. traditionnels c'est-à-dire appliqués principalement aux problèmes administratifs et de gestion courante des entreprises, de nombreuses méthodes existent (IDA, MERISE, REMORA, Booch...) pour aider la conception et la réalisation d'un S.I.. De plus l'analyste a souvent un grand choix d'A.G.L. (Atelier de génie logiciel nommé en anglais par CASE computer aided software engineering) pour l'aider dans sa conception (par exemple : DBmain).

Par contre pour les S.I.G. il n'existe que très peu de véritables méthodes et d'outils "CASE" spécifiques à leur développement.

Dans un premier temps les S.I.G. étaient principalement considérés comme une boîte à outils, il ne semblait donc pas nécessaire d'utiliser une méthode de développement pour les implanter dans une organisation. D'autant plus que bien souvent les vendeurs de S.I.G.-logiciels font croire aux utilisateurs que leur achat suffit pour créer un système informatisé.

Ensuite, dans les années 80, quand la notion du S.I.G. a dépassé celle du simple logiciel et qu'il a été considéré comme un véritable S.I. on a cherché à appliquer les méthodes de conception des S.I. dits traditionnels. Très vite beaucoup de chercheurs se demandèrent alors s'il fallait créer une méthode particulière pour développer un S.I.G..

Enfin depuis quelques années plusieurs personnes pensent que vu la difficulté d'appliquer les méthodes traditionnelles et les éléments particuliers des S.I.G. informatisés, il faut déterminer une méthode spécifique à leur conception.

Ces éléments particuliers nous les avons mis en évidence dans le chapitre 1. Ce sont principalement les caractéristiques des données géo-graphiques et la manière particulière dont les quatre fonctions de base (acquisition, gestion, transformation / analyse, présentation / diffusion de l'information) sont réalisées par les S.I.G.. Rappelons aussi qu'une des différences majeures est que l'informatisation d'un S.I.G. d'une organisation s'effectue en grosse partie par l'intégration dans l'organisation d'un S.I.G.-logiciel. Signalons donc que choisir et acheter un S.I.G.-logiciel est une étape (en général appelée benchmark) très importante dans le développement d'un S.I.G.. D'autant plus que comme le signale Pantazis il "peut influencer , non seulement la réalisation / implantation du système mais aussi la manière dont il sera conçu!" [Pantazis 94, p.209]

Actuellement, à notre connaissance, rares sont les véritables propositions de méthodologie de conception de S.I.G.. En effet la plupart des méthodes exposées à ce jour ne sont pas complètes : il manque presque toujours au moins un des composants de la méthode rappelée ci-dessus.

Pour illustrer ces propos, nous pouvons reprendre la conclusion de l'article "Quelques réflexions sur la difficulté d'utiliser MERISE pour la modélisation des bases de données géographiques " du consultant Pornon.

"MERISE est-elle la méthode de modélisation des données la plus adaptée au cas de l'information géographique? Difficile de répondre à cette question. La méthode a de nombreux avantages : diffusion importante en France, évolution vers les concepts objets qui corrige certains de ces défauts, rigueur de l'approche qui, malgré les inconvénients signalés, lui confère une grande puissance, existence d'AGL à des prix raisonnables [...]. Ses inconvénients sont d'une part, les lacunes de la démarche traitement, d'autre part son approche trop orientée gestion. Des démarches [...] cherchant des adaptations sont très appréciables mais doivent nécessairement aller au-delà du simple formalisme de représentation pour couvrir tout le cycle de modélisation des données et aboutir au développement d'AGL adaptés au cas des S.I.G.."[Pornon 93, p.262]

Toutefois signalons le remarquable travail qui se réalise au Québec et particulièrement à l'université de Laval au centre de recherche en géomatique. Ils sont notamment très actifs depuis quelques années dans la recherche concernant les aspects méthodologiques de la conception des S.I.G. Par exemple la méthodologie élaborée au centre de recherche de Laval est une véritable méthodologie avec une démarche fondée sur des modèles et exploitée par des outils "CASE". Ils inventèrent notamment pour cette méthodologie un modèle de structuration de l'information, un des plus connus actuellement dans le monde de la géomatique. Ce modèle se nomme MODUL-R et est une adaptation pour l'information géographique du modèle E-R. Un reproche que l'on peut faire à la méthodologie et souligné par Pantazis est qu'elle ne tient pas assez compte de l'organisation dans laquelle est réalisé le projet.

Dans sa thèse de doctorat, Pantazis a réalisé une véritable méthode de conception des S.I.G. appelée MECOSIG pour laquelle son modèle CONGOO, que nous allons étudier en détails dans la suite, a été développé. Nous ne présentons pas cette méthode puisque ce n'est pas le but de notre travail et qu'elle ne sera entièrement disponible au publique dans le livre "MECOSIG et CONGOO, méthode et formalisme de conception de S.I.G." qui paraîtra dans les prochains mois. Nous renvoyons donc au livre tout lecteur intéressé par plus de renseignements sur la méthode.

Signalons tout de même que d'après ce que nous savons [Pantazis 94a], cette méthode est composée :

- de "principes" c'est-à-dire d'une base théorique et d'un fil conducteur des concepts proposés.
- d'une démarche multiple découpée en différentes étapes et "noeuds de décisions" qui déterminent l'acheminement du projet selon les choix effectués par le concepteur et déterminés par des contraintes et les conditions imposées par la réalité ainsi que les souhaits de l'utilisateur.
- d'un ensemble de modèles. Rappelons à ce propos que c'est pour cette méthode que le modèle CONGOO a été inventé. Nous l'étudions dans la partie suivante.

Notons que pour le moment aucun outil "CASE" n'a été conçu pour cette méthode mais des recherches sont en cours pour l'instant.

2.3. Les modèles conceptuels de structuration de l'information géographique

Afin d'étudier en détails le modèle de structuration de l'information géographique CONGOO, nous rappelons tout d'abord ce qu'on entend par modèle de structuration des informations au niveau conceptuel et ces objectifs. Ensuite nous exposons les problèmes de modélisation de l'information géographique avec le modèle E-A et discutons les adaptations trouvées dans la littérature pour résoudre ceux-ci. Enfin le modèle CONGOO est présenté et critiqué en détails (partie 2.4).

2.3.1. Objectifs des modèles de structuration de l'information

Comme pour la conception des S.I. traditionnels, modéliser les données est une étape fondamentale dans la conception des S.I.G..

Pour la conception des S.I., depuis de nombreuses années, on utilise différents niveaux de modélisation des données. En partant du monde réel (correspondant au sous-ensemble de la réalité qui nous intéresse), trois principaux niveaux sont généralement distingués :

- Le niveau conceptuel (description de concepts indépendamment d'une informatisation future)
- Le niveau logique (où l'on prend en compte les caractéristiques des S.G.B.D. utilisés)
- Le niveau physique (implantation physique de la base de données)

L'utilité de ces différents niveaux est très bien admis par les concepteurs de S.I.G.. Toutefois il y a une différence essentielle que nous avons déjà signalée à plusieurs reprises avant, l'informatisation d'un S.I.G. se fait principalement par un S.I.G.-logiciel ou plutôt progiciel, il n'est donc pas utile de descendre au niveau physique car "[...]le niveau physique est directement offert aux utilisateurs. Ainsi, la conception /réalisation d'une base de données géographiques peut s'arrêter au niveau logique (ou plus correctement au niveau logique/physique)" [Pantazis 94, p.241].

Rappelons les objectifs d'un modèle de structuration des informations de niveau conceptuel.

- Aider les analystes à exprimer la sémantique des données contenues dans la mémoire du S.I. à l'exclusion des problèmes de représentation physique des données, d'accès aux données et d'organisation du stockage et ceci en adéquation avec les besoins du client.

- Doit servir de base de communication entre les différentes personnes intéressées au développement du S.I..

"- Aider les réalisateurs du S.I. (informaticiens, programmeurs) à comprendre cette sémantique en vue d'organiser un stockage adéquat des données et de créer des procédures d'accès efficaces ainsi que des programmes d'application corrects.
[...]
- Procurer aux exploitants du S.I. la définition précise des informations qu'ils manipulent ainsi que leurs conditions d'utilisation.
[...]
- Fournir une représentation opérationnelle des bases de données du S.I. directement exploitables à l'aide de langages d'interrogation de haut niveau [...] et de langages dits de 4ème génération." [Bodart et Pigneur 93, p.14]

Pour atteindre ces différents buts le modèle doit disposer des caractéristiques suivantes :

- Une grande lisibilité et une compréhension facile. Le modèle étant la base de communication entre les personnes intéressées par le développement du S.I., il est important qu'il soit le plus lisible possible. Il est évident que les personnes qui le lisent et l'utilisent doivent maîtriser parfaitement le modèle. Il est donc important que celui-ci soit facilement compréhensible par toutes les personnes(programmeur, analyste, client) impliquées dans le développement.

- Une sémantique clairement définie non contradictoire et non ambiguë. Puisque le modèle doit servir de base de communication entre différentes personnes, il est important que les concepts du modèle aient une interprétation unique et précise. Par exemple les langages naturels souffrent d'un manque évident de précision, il est donc fréquent qu'un même texte soit compris de façons différentes selon les lecteurs. Il est important qu'un langage possède une syntaxe formelle et une interprétation formelle complète qui garantissent l'absence d'ambiguïté. De plus les langages formels munis de règles de déduction permettent de concevoir des outils logiciels performants qui peuvent vérifier la complétude et la cohérence d'un schéma.

- Une grande expressivité. C'est-à-dire que le modèle doit supporter facilement le "mapping" et si possible un "mapping" naturel, entre toutes les choses de la partie du monde "réel" devant être décrites dans la mémoire S.I. et les différents concepts à disposition dans le langage.

- Posséder des mécanismes de structuration afin de pouvoir manipuler et maintenir la lisibilité d'une masse considérable d'informations.

2.3.2. Les lacunes et les adaptations des modèles traditionnels pour structurer l'information géographique

Dans cette partie nous exposons les difficultés à modéliser les données géographiques avec les modèles utilisés pour structurer les données des S.I. "traditionnels". Nous étudions en fait juste le modèle E-A (Entité-Association) mais les remarques que nous faisons à propos de ce modèle sont facilement transposables aux autres modèles. Le modèle E-A permet d'exprimer la sémantique des données mémorisables à l'aide des concepts d'entité, d'association, d'attribut et du mécanisme des contraintes d'intégrité. Ce modèle a une très grande audience parmi les spécialistes des S.I. et il est reconnu que l'utilisation de celui-ci permet de réaliser des schémas de grande qualité de communication et de bonne capacité de représentation des informations appartenant au réel perçu. Nous n'exposons pas en détails les concepts du modèle E-A dans ce travail et renvoyons toutes les personnes ne les connaissant pas aux nombreux ouvrages qui les expliquent comme par exemple [Bodart et Pigneur 93], [Chen 76],...

Le modèle E-A est souvent utilisé par les concepteurs de S.I.G. pour modéliser l'information géographique. La plupart ont souligné les lacunes de celui-ci et ont souvent effectué des adaptations.

Pour illustrer ces propos, observons l'exemple ci-dessous de la modélisation de l'information géographique avec le modèle E-A.

Exemple

Cet exemple est en partie tiré de l'énoncé de l'exercice de modélisation donné au cours postgrade par Caron. [Bédart et Caron 93].

Phrases types :

1) La commune souhaite connaître toutes les parcelles de son territoire et les représenter sous forme polygonale.

2) Chaque parcelle est identifiée de façon unique à la fois par son numéro et par le numéro de la commune où elle est située.

3) On gère aussi les bâtiments dans la commune, ils sont identifiés par leur numéro d'assurance.

4) Chaque bâtiment est représenté par une surface.

5) Pour chaque bâtiment, on note aussi son numéro de police, son nom et son genre (résidentiel, commercial, industriel..).

6) Un bâtiment est construit sur au moins une parcelle.

7) Une parcelle peut n'avoir aucun bâtiment, sinon en avoir plusieurs.

8) Pour chaque parcelle, on veut conserver le numéro de plan où elle figure, la superficie de la parcelle, le genre de propriété de même que la date de son entrée en vigueur.

9) Pour chaque parcelle , on souhaite connaître son propriétaire dont on conserve le prénom, le nom et la date de naissance.

10) Alors qu'un propriétaire possède au moins une parcelle sinon plusieurs, de la même manière une parcelle est possédée par au moins un propriétaire.

11) Un propriétaire habite dans un et un seul bâtiment.

12) La commune veut gérer les arbres se trouvant sur les parcelles, ceux-ci doivent être représentés par un point.

13) Chaque arbre a un numéro l'identifiant, sa taille et son type.

14) Un arbre se trouve sur une et une seule parcelle.

15) Une parcelle peut n'avoir aucun arbre, sinon en avoir plusieurs.

16) Dans un bâtiment peut vivre aucun propriétaire, sinon plusieurs.

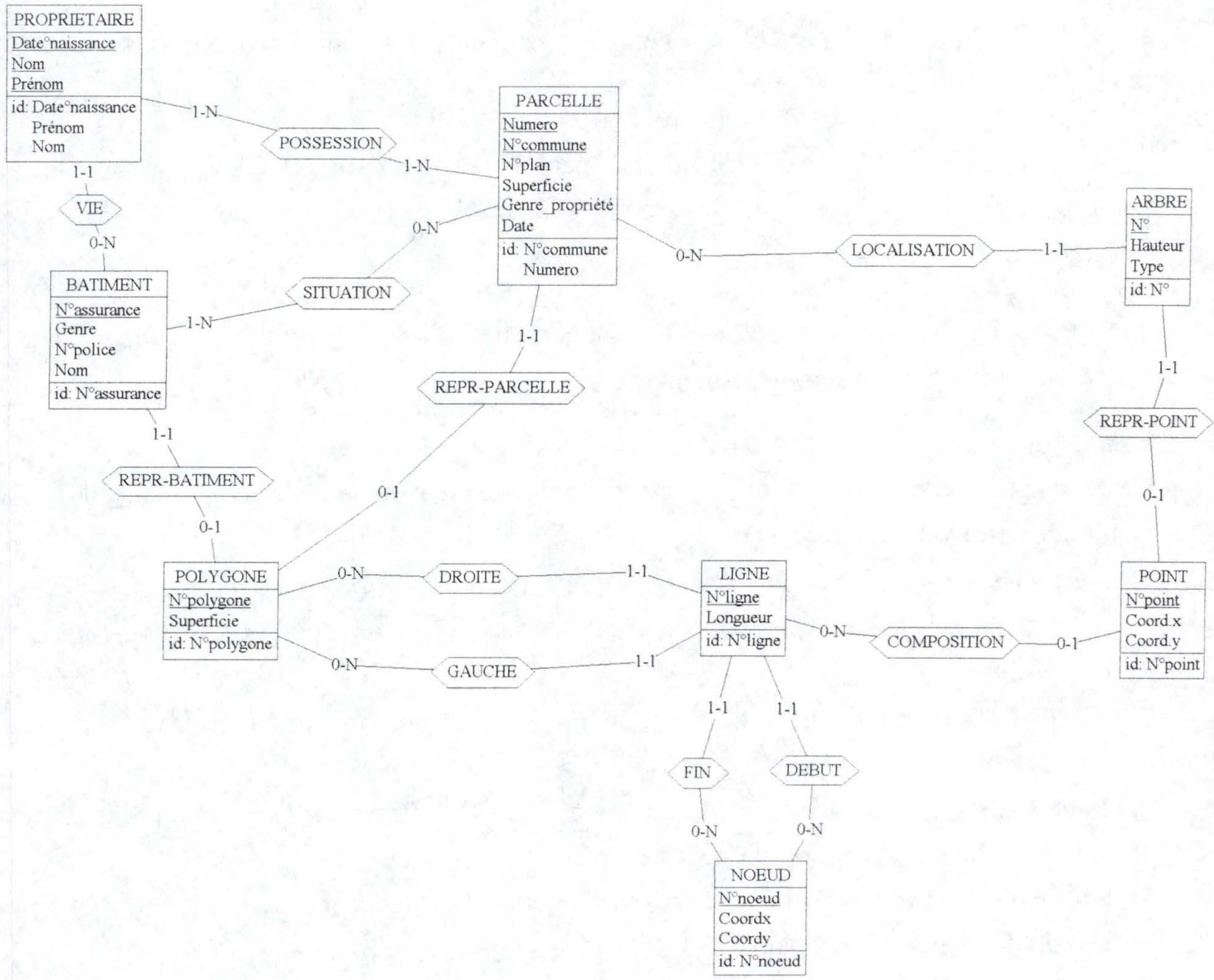


Figure 2.2 : Exemple de schéma conceptuel contenant des entités géométriques

Dans notre exemple nous avons modélisé la référence spatiale des entités comme traditionnellement c'est-à-dire en ajoutant des entités géométriques dans le modèle conceptuel de données de base.

Cette manière de procéder, comme l'a très bien souligné Caron [Caron, Bédart et Gagnon 93], a plusieurs inconvénients majeurs.

Tout d'abord le fait d'incorporer au schéma conceptuel de données, des entités géométriques pour exprimer la référence spatiale rend le schéma plus difficile à lire et à communiquer. Ceci est d'autant plus facile à comprendre quand on sait qu'en général un schéma conceptuel de données géographiques a parfois plusieurs centaines de types d'entités à références spatiales qui devraient être tous rattachés à un des trois types entités géométriques. Cela serait totalement illisible et peu compréhensible, or ce sont là deux des qualités les plus importantes demandées à un schéma conceptuel.

Ensuite le fait de trouver dans un même schéma à la fois des entités réelles (Bâtiment, arbre, parcelle,..) et des entités géométriques (Point, ligne, polygone,...) apparaît un peu incohérent.

Enfin les S.I.G.-logiciels, servant à l'informatisation des S.I.G., possèdent leur propre structure interne de gestion des données géométriques. Celle-ci étant souvent peu documentée, il est donc difficile pour l'utilisateur de décrire précisément cette structure géométrique au niveau conceptuel. De plus il n'y aurait plus indépendance de la solution donnée par le schéma conceptuel des moyens de réalisation, l'objectif premier de la réalisation d'un schéma conceptuel. Cette structure géométrique, au niveau conceptuel, est surtout intéressante pour le développeur d'un S.I.G.-logiciel et très peu pour le concepteur d'une base de données utilisant un S.I.G.-logiciel.

Plusieurs solutions ont été apportées pour garder les formes géométriques associées aux entités réelles sans alourdir le schéma conceptuel. Il ne suffisait pas de supprimer les entités géométriques pour résoudre les problèmes car il est important de garder les formes géométriques associées aux entités réelles dans le but de préciser quelles sont les entités qui ont une représentation spatiale, afin de pouvoir indiquer au niveau conceptuel avec quelles formes géométriques seront numérisées les diverses entités dans le S.I.G. et pour faciliter la compréhension des relations spatiales entre entités.

La solution la plus intéressante consiste à associer à l'entité qui a une description géométrique, un pictogramme spatial indiquant la représentation géométrique des entités c'est-à-dire préciser s'il s'agit de point, ligne, polygone ou de combinaison de ces éléments. Cette solution a l'avantage d'être relativement intuitive pour le lecteur et est très facile à utiliser.

Soulignons que les pictogrammes spatiaux indiquent la forme géométrique choisie par l'utilisateur pour représenter l'entité dans la base de données. Cette forme peut être différente de la forme réelle de l'objet . Par exemple une rue peut être vue comme une surface ou une ligne suivant les besoins de l'utilisateur (voir 1.2 au chapitre 1).

Le modèle MODUL-R [Caron, Bédart et Gagnon 93] considéré comme un des modèles les mieux adaptés pour modéliser l'information géographique a adopté cette solution. Le modèle MODUL-R est basé sur le modèle E-R, auquel il ajoute trois modules complémentaires. Ces modules regroupent diverses techniques de modélisation. Selon les modules qu'il décide d'utiliser, un modélisateur peut ainsi créer un schéma conceptuel de données avec seulement le modèle individuel de base sinon en combinant divers ajouts intégrés dans MODUL-R. Un des sous-module est le module de sous-référence spatiale où les pictogrammes spatiaux sont utilisés pour spécifier si la représentation géométrique d'une entité est de type point, ligne, surface ou volume.

Reprenons les règles de MODUL-R concernant les pictogrammes spatiaux (Figure 2.3)à titre d'exemple.

"Un pictogramme spatial signifie "présence de" cette forme pour représenter l'entité. Ainsi, un pictogramme de type point indique que toute occurrence de l'entité sera représentée par au moins un élément géométrique de type point.[...] Lorsqu'il n'y a pas de pictogramme de référence spatiale dans une entité, cela signifie que l'on ne désire pas la cartographier."

"Un pictogramme spatial peut contenir plus d'une forme géométrique : on parle alors d'un pictogramme spatial complexe. Ce type de pictogramme signifie que pour chaque occurrence d'un type d'entité, une agrégation de ces formes géométriques est nécessaire pour la représenter."

"On peut aussi utiliser un pictogramme spatial alternatif, indiquant qu'une occurrence d'entité peut avoir l'une ou l'autre des représentations géométriques du pictogramme, mais pas les deux." [Caron, Bédart et Gagnon 93, p.290]

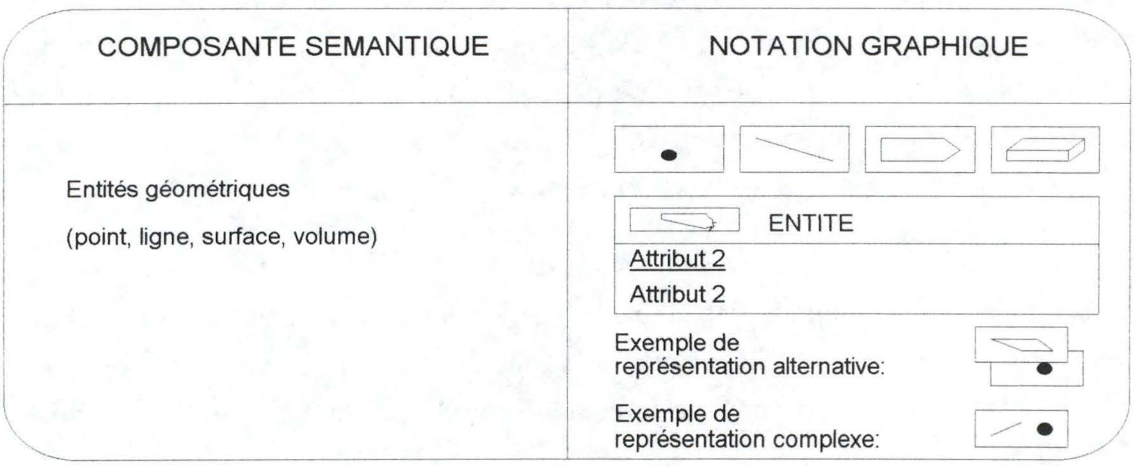


Figure 2.3 : Pictogrammes de référence spatiale représentant des entités géométriques [Pageau, Bédart et Gagnon 93]

Si nous effectuons à présent la modélisation, de l'exemple vu plus haut, en utilisant ces règles nous voyons que le schéma est beaucoup plus lisible.(Figure 2.4)

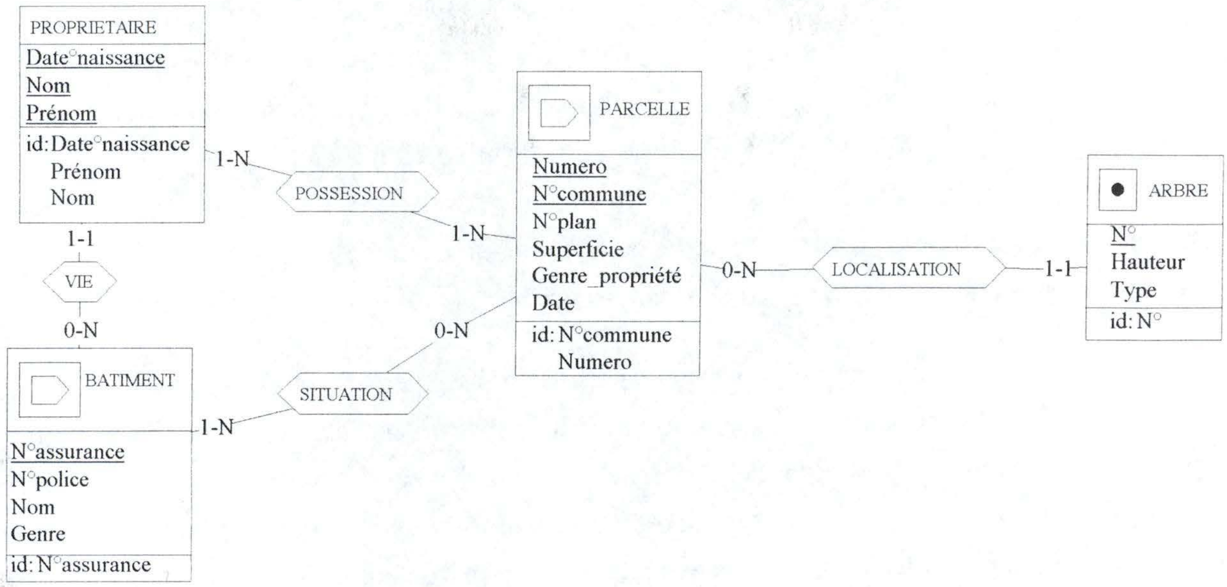


Figure 2.4 : Exemple de schéma conceptuel avec des pictogrammes spatiaux

Signalons qu'une autre solution peut être trouvée dans la littérature. Elle est basée sur le constat que le plus important dans un schéma conceptuel de données n'est pas la géométrie mais la sémantique. Cette solution consiste à rendre compte de la géométrie dans l'entité mais sous forme d'attribut dont son domaine est un nouveau type de donnée géométrique (tel que point, ligne, surface) ajouté au type de donnée standard (string, integer,...). Cette solution a l'avantage de pouvoir utiliser certains AGL qui permettent de définir des types utilisateurs.

D'autres reproches sont faits au modèle E-A pour modéliser les données géographiques et de nouvelles adaptations sont nécessaires.

Dans le domaine des S.I.G., les bases de données sont souvent très complexes, c'est-à-dire qu'elles ont en général un très grand nombre d'entités, de relations et d'attributs. La lecture, la compréhension, la manipulation d'un schéma E-A se fait donc difficilement.

De plus on reproche aussi au modèle E-A son manque de richesse d'expression qui ne permet pas une traduction facile du schéma pour les outils orientés-objets de plus en plus nombreux (par exemples les SGBD Orienté Objet). Au sujet de l'orienté objet, notons que beaucoup de chercheurs montrent l'intérêt des modèles orientés objets pour la conception de bases de données géographique.

Pour résoudre les problèmes de complexité et de manque de richesse d'expression, beaucoup de chercheurs ont ajouté des mécanismes qui permettent de gérer la complexité et/ou d'augmenter la richesse d'expression du modèle E-A.

Un des mécanismes les plus répandus dans les modèles adaptés pour la modélisation des données géographiques est le mécanisme d'héritage qui est introduit à l'aide des concepts de spécialisation et de généralisation. Ce mécanisme a différentes règles suivant les modèles (GéO2 [David et al 93], MODUL-R, IFO [Abiteboul et Hull 87],.[Worboys et al 90].)dans lesquels il est utilisé mais les grands principes restent les mêmes : la spécialisation permet de raffiner un sous-type à partir d'un type déjà défini, à l'inverse la généralisation permet de regrouper plusieurs types ayant des propriétés communes.

Un autre mécanisme que l'on trouve parfois, principalement pour faciliter la traduction du modèle vers la modélisation orienté-objet, est le mécanisme d'agrégation. Ce mécanisme permet la création d'entités complexes. Celui-ci réunit des entités par la relation "se compose de" (ou partie de) où certaines contraintes d'agrégation peuvent s'appliquer. Signalons qu'il existe plusieurs mécanismes similaires avec des règles légèrement différentes et parfois un nom différent (mécanisme de propagation, mécanisme de composition).

Nous pouvons aussi trouver le mécanisme de groupement. Ce mécanisme groupe un ensemble d'objets d'un même type d'entité pour former un objet d'un type d'entité de plus haut niveau.

Pour gérer les grandes applications et faciliter la lecture du schéma, MODUL-R propose le regroupement des entités par thèmes. "La création d'un thème peut être justifiée, soit par une logique sémantique, soit par une grande interrelation d'entités entre elles, soit par un regroupement physique des données, soit par l'utilisation des données par un même groupe d'utilisateurs. Par exemple : circulation, loisirs et infrastructure peuvent être considérés comme différents thèmes concernant une ville."[Caron, Bédard et Gagnon 93, p.295]

MODUL-R propose aussi cinq niveaux de détails qui permettent de voir un même modèle à un niveau très sommaire jusqu'à un niveau très détaillée.

Notons aussi que souvent, outre la prise en compte du contexte de la référence spatiale, il faut modéliser la référence temporelle. Celle-ci demeure difficilement représentable de manière simple avec le modèle E-A. De nouveau plusieurs adaptations et un nouveau modèle ont été créés pour améliorer la lisibilité du schéma.

Quelques remarques sur les particularités du processus de modélisation de l'information géographique

Nous avons donné ci-dessus (Figure 2.4) un exemple de modélisation de données géographiques. Cet exemple est basé sur une partie d'un exercice donné par Caron à ses étudiants. L'énoncé de l'exercice est constitué d'un ensemble de phrases types assez simples à modéliser. Dans la réalité, la modélisation de l'information géographique est bien plus difficile à réaliser parce qu'elle est basée sur différents types de documents. En effet l'analyste qui doit structurer les données d'un S.I.G. est obligé d'examiner une multitude de types de documents tels que cartes, plans, photos-aériennes, textes...

Le processus d'identification des types d'entités et de leurs attributs à partir de documents textuels tels que textes, factures, bons de commande et autres formulaires est bien connu grâce à l'informatique de gestion. Par contre le processus d'analyse et de traduction, sous forme de données de base, de l'information contenue dans un plan, une carte, une photo-aérienne est fort mal connu et n'a jamais fait l'objet d'une étude précise. Ce processus reste donc totalement empirique.

Signalons que le choix des données de base est beaucoup plus difficile à faire pour un S.I.G. que pour un S.I. "traditionnel" car contrairement à celui-ci, il est rarement possible de définir a priori tous les traitements qui seront effectués par l'utilisateur. En effet comme le dit très bien Pornon : "On peut en général, en informatique de gestion, prévoir à l'avance tout ce qui va se passer et donc toutes les données nécessaires. Les processus de traitement des factures et de gestion des stocks peuvent être analysés ou définis a priori et il s'agit essentiellement de les rendre cohérents.[...] Certaines applications des S.I.G. se rapprochent de cette vision de l'informatique : la gestion d'un réseau d'eau ou d'assainissement, celle des procédures d'urbanisme sont codifiables et leur processus est prévisible. En revanche, d'autres applications concernent des processus et des choix sur les données qu'on ne peut prévoir à l'avance. C'est le cas des utilisations des S.I.G. à fins d'analyse spatiale." [Pornon 93, p.257]

Quand tous les types d'entités et d'attributs sont identifiés, les difficultés ne sont pas terminées pour l'analyste. Le choix et l'identification des relations à représenter entre types d'entités est encore un problème plus complexe à résoudre. En effet, quand on doit représenter les relations entre types d'entités, beaucoup de questions nous viennent à l'esprit auxquelles il est très difficile de répondre et que la littérature scientifique n'a jamais abordés précisément à notre connaissance. "Faut-il représenter les relations spatiales (inclusion par exemple) : toutes, aucune, seulement les pertinentes ?" [Pornon 93, p.259]. Comment les identifier ? Que faut-il faire des règles qui concernent l'intégrité géométrique ou topologique des données ?

Par exemple dans l'exercice vu plus haut (Figure 2.4), faut-il modéliser le fait qu'un arbre ne peut pas se trouver à l'emplacement d'un bâtiment? Répondre à toutes ces questions est d'autant plus difficile que, comme nous l'avons vu plus haut, on ne sait pas prédire tous les traitements que l'on va exécuter.

Prenons un autre exemple de modélisation basé sur des cartes du Grand-Duché du Luxembourg (Figure 2.5). Admettons que le client demande que les différentes zones administratives soient représentées graphiquement et que des données alphanumériques telles que le nom, le nombre d'habitants, la superficie soient reliés à chaque zone.

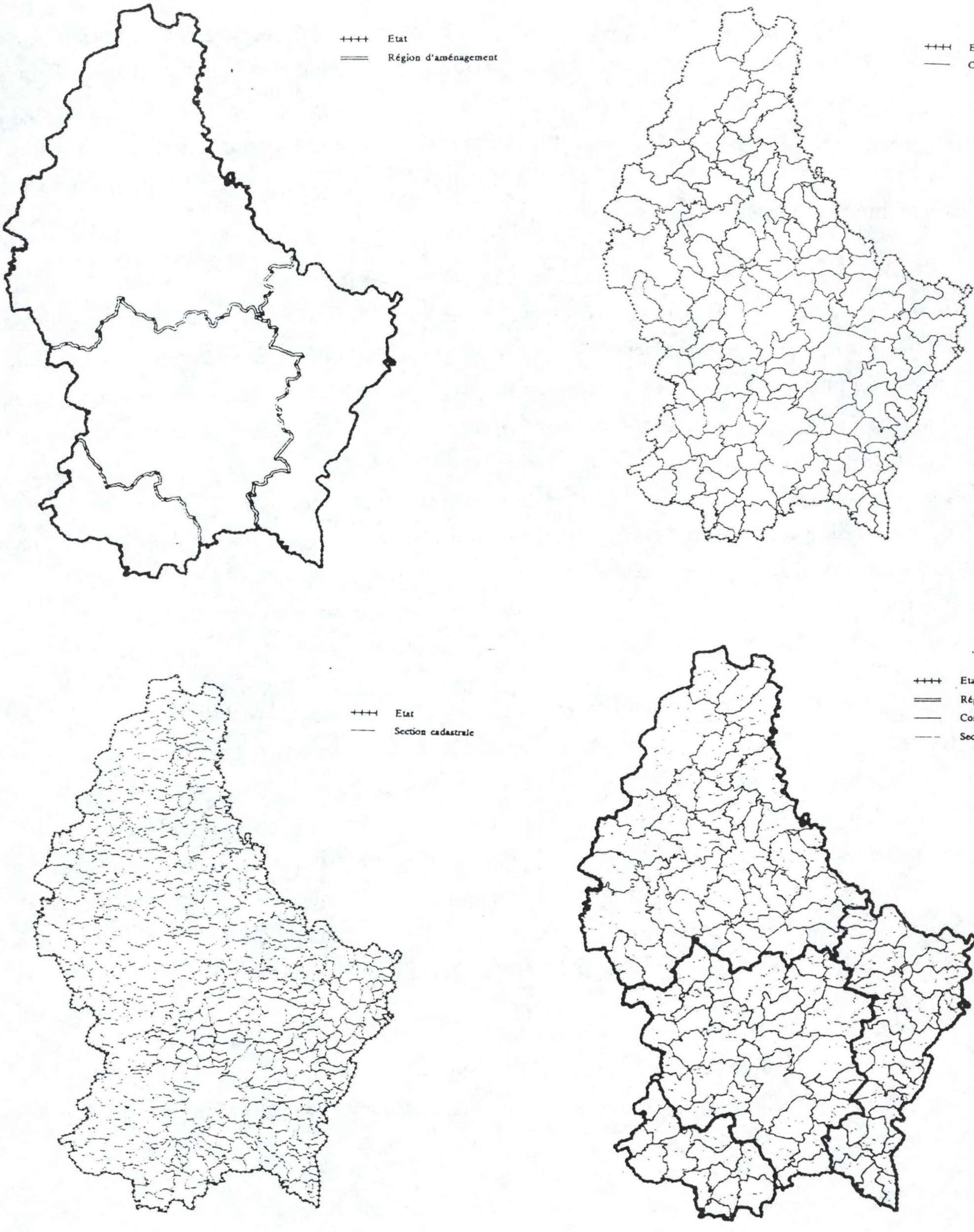


Figure 2.5 : Les quatre cartes à modéliser

Sur cet exemple, les types d'entités sont faciles à identifier ainsi que leurs attributs. Par contre, le choix des relations spatiales qu'il faut représenter est beaucoup plus difficile à faire. Le schéma ci-dessous est il suffisant? (Figure 2.6)

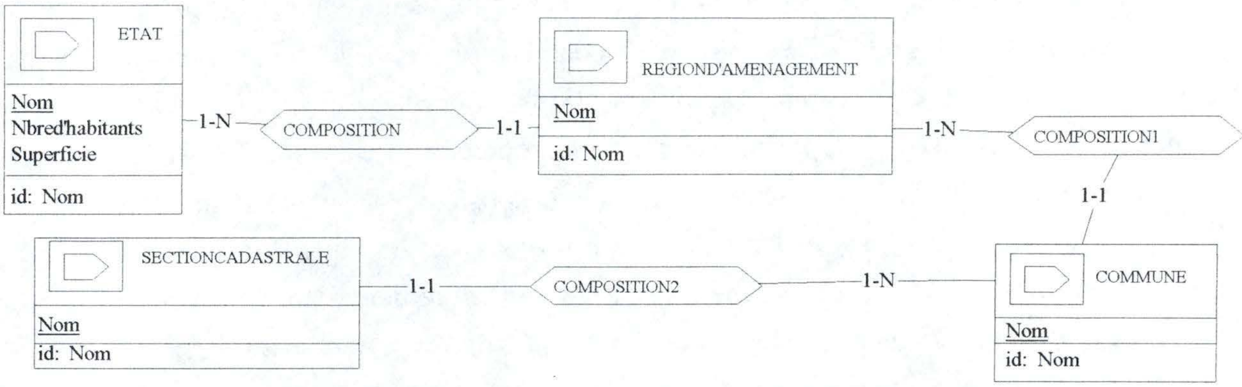


Figure 2.6 La modélisation des cartes du Grand-Duché du Luxembourg

Bien que nous n'ayons pas une grande expérience dans la modélisation de données géographiques et la conception de S.I.G., nous tentons ci-dessous de donner quelques éléments de réponse.

Tout d'abord, il nous semble normal de modéliser les relations spatiales qui interviennent dans la définition d'un type d'entité.

Ensuite, celles qui sont explicitement énoncées par le client doivent aussi être modélisées.

Enfin, pour avoir une base de données cohérente, il faut exprimer toutes les contraintes d'intégrité et notamment celles qui incluent la position des objets c'est-à-dire les contraintes d'intégrité topologique. En effet dans un S.I.G. la topologie est à prendre en considération pour déterminer si l'état d'une base de données est correcte ou non. Par exemple dans la figure 2.4, le fait qu'un arbre ne peut pas être sur le même emplacement qu'un bâtiment est intéressant à modéliser pour contrôler la cohérence de la base de données. Les contraintes topologiques nous semblent aussi importantes pour donner des indications au digitalisateur de cartes, de plans,...

Signalons que, si toutes les relations topologiques entre types d'entités sont modélisées, cela crée une redondance, seules quelques relations devraient être stockées mais en général pour des raisons d'efficacité elles sont toutes stockées. Nous reviendrons sur ce problème dans le chapitre quatre consacré aux relations topologiques.

Maintenant, d'après nos réponses, le schéma de la figure 2.6 ne semble pas complet car on n'a pas représenté les contraintes topologiques telles que deux communes ne peuvent être superposées, l'ensemble des communes d'un pays occupe le même espace que le pays,....

Si nous voulions représenter graphiquement toutes ces contraintes topologiques avec le modèle MODUL-R le schéma deviendrait fort peu lisible. Nous allons étudier dans la section suivante un modèle qui permet de représenter facilement les contraintes topologiques.

A propos des relations topologiques, il nous semble aussi important de signaler qu'une multitude de termes sont utilisés pour définir la relation entre des objets (inclure, partager, toucher, être adjacent, appartient, intersection, connecté, inclusion, égalité, couvrir, contenir,...). Ceux-ci ont rarement une sémantique précise, il est donc souvent interprété de manières différentes par les lecteurs. Or comme nous l'avons vu dans l'introduction de ce travail, il est important d'avoir une définition claire et précise pour que le développement du S.I.G. soit mené à bien. Il nous semble donc primordial de définir formellement les relations topologiques d'autant plus qu'elles jouent un rôle très important dans le S.I.G. (relation, contrainte d'intégrité, interrogation de la B.D.). Nous reviendrons sur ce problème dans le chapitres 3 et 4.

2.4. Présentation du modèle CONGOO

Le modèle CON.G.O.O. (CONception Géographique Orientée Objet) a été proposé dans le cadre de la création d'une méthode de conception pour système d'information géographique appelée MECOSIG. Cette méthode et ce modèle sont les résultats des recherches effectuées par Pantazis dans sa thèse de doctorat en sciences à l'université de Liège au laboratoire SURFACE (obtenue en décembre 94). Ce modèle concerne principalement les "données géographiques" ainsi que les données alphanumériques attributaires ou non à ces données.

Depuis la fin du doctorat le modèle a continué à évoluer rapidement et de nombreuses recherches sont encore en cours pour l'améliorer. Pour notre présentation nous nous basons sur le chapitre 5 "le formalisme CONGOO" du livre "MECOSIG et CONGOO, méthode et formalisme de conception de S.I.G." qui va paraître prochainement.

Pantazis a créé ce modèle parce qu'il estimait que les modèles utilisés jusqu'à présent pour structurer l'information géographique avaient plusieurs lacunes importantes :

" - distinction insuffisante des formalismes existants à propos des entités (objets) et de "groupes"(classes) de ces entités.
- représentation inappropriée des relations topologiques.
- distinction lacunaire des relations topologiques avec d'autres types de relations.
- représentation partielle des contraintes topologiques entre les entités (objets) représentées.
- Sous-intégration des modèles des traitements et des données."[Pantazis 94a, p.1308]

Avant de présenter ce modèle, il nous a semblé important de définir plusieurs termes utilisés très fréquemment dans le domaine de l'informatique, souvent avec des significations différentes d'un auteur à l'autre.

"Objet : une abstraction de quelque chose du domaine du problème, reflètant les capacités d'un système à garder de l'information sur elle, d'interagir avec elle ou de faire les deux choses à la fois; une encapsulation des valeurs des attributs et des services qui leur sont spécifiques (synonyme instance)".[Coad et Yourdon 91]

"Type : structure commune aux éléments d'une classe"[Hainaut 94]

"Classe : ensemble d'instances qui jouissent de propriétés structurelles identiques".[Hainaut 94]

Ces significations sont celles qui nous semblent les plus admises. Seulement Pantazis utilise dans son modèle , ces termes mais avec une autre signification. Nous ferons donc en temps voulu quand il sera possible la correspondance avec "nos" termes. Passons à présent à la présentation succincte des concepts du modèle CONGOO.

2.4.1. Présentation succincte des concepts de CONGOO

2.4.1.1. les objets

Le concept central du modèle est l'objet. La définition de l'objet est :

"R.1. Un objet peut être une abstraction de toute chose, personne, événement ou concept ayant ou non une référence spatiale déterminée.[...]L'objet est la représentation individuelle de chaque occurrence reliée sémantiquement avec cet objet.[...]la création d'un objet signifie pratiquement créer une fiche vide, avec une structure donnée. La création d'instances correspondant à l'objet revient a remplir cette fiche.[...]"

Remarquons que cette définition d'objet semble correspondre à un type pour nous.

Chaque objet est composé d'un nom, d'un pictogramme (point, ligne, polygone,...) ou d'une mention textuelle de la catégorie et du type d'objet (OGS-P, OGS-L, ..., l'inscription alpha sera indiqué si l'objet n'a pas de représentation graphique), d'un identifiant, de plusieurs attributs logiques (données concernant l'identité et les caractéristiques de l'objet du monde réel qu'il représente) et des attributs graphiques (données qui concernent sa "représentation / visualisation") pour les objets géo-graphiques. Plusieurs traitements peuvent être associés à chaque objet.

Différentes catégories d'objets sont définies. Tout d'abord les objets géo-graphiques sont distingués des objets non géographiques.

Les objets géo-graphiques sont des objets à référence spatiale et représentés graphiquement en deux dimensions par des points, lignes, polygones ou une combinaison de ces éléments.

Les objets non géo-graphique sont des objets avec ou sans référence spatiale mais sans représentation graphique.

La catégorie des objets géo-graphiques est divisée en trois catégories d'objets chacune divisée en sous-catégories.

1. Les objets géo-graphiques simples (OGS) sont des objets dont le mode d'implantation prend l'une des quatre formes suivantes : point, ligne, polygone, surface. Ainsi il existe quatre sortes d'objets simples : les OGS du type point (OGS-P), du type ligne(OGS-L), du type polygone (OGS-Pol), du type surface (OGS-S).

2. Les objets géo-graphiques composés (OGC) sont des objets qui proviennent de la réunion ou de la division des objets simples de même mode d'implantation. Il existe aussi trois sous-catégories d'objets composés (type point, ligne, polygone).

3. Les objets géo-graphiques complexes (OGX) sont des objets composés qui regroupent différents types d'objets (parmi les trois : points, ligne, polygone) simples ou composés.

2.4.1.2. Les classes

Cette notion ne doit pas être confondue avec la notion de classe que l'on a l'habitude de voir en informatique mais est définie :

"R.43 Une classe couvre l'ensemble des objets sémantiquement homogènes qu'elle contient (population d'objets), et regroupe l'ensemble des instances représentées par ces objets. Une classe peut inclure aussi (sans que cela soit obligatoire) les contraintes topologiques qui s'appliquent entre toutes les instances représentées par les objets appartenant à la classe, ainsi que les traitements qui s'appliquent aux objets eux-mêmes. Des attributs, nommés attributs de la classe et se rapportant à la population des instances de la classe, peuvent être aussi mentionnés (par exemple, le type de la distribution spatiale des instances de la classe)" .

Ce concept nous semble fort intéressant pour modéliser les données géo-graphiques car il permet de décrire la distribution et les relations topologiques des instances des objets contenus dans une classe. Signalons qu'un objet est toujours associé a une classe.

2.4.1.3. Les couches

"[...]le terme «couche» est défini pour refléter des ensembles d'objets et des classes d'objets ayant des relations en commun. Les couches sont donc des composants utilisés pour communiquer / décrire la totalité de la base de données d'un système global, ainsi que l'ensemble des relations, notamment topologiques, communes entre les objets et/ou classes appartenant à la même couche."

"R.58. La création d'une couche doit être basée non pas sur l'homogénéité sémantique des objets mais, surtout, sur les relations qu'ils partagent entre eux."

2.4.1.4. Les sous-couches

"R.63. Les sous-couches constituent des ensembles de classes d'objets appartenant à la même couche. Ces classes sont rassemblées en une sous-couche en raison des relations, notamment topologiques, qu'elles entretiennent entre elles, ou que les objets ou les instances qu'elles contiennent entretiennent entre eux."

2.4.1.5. Les relations

" R.70. Le formalisme CONGOO distingue trois types de relations : les relations de structure, les relations topologiques et les relations logiques."

2.4.1.5.1. Les relations de structure

"R.72. Il y a deux différentes relations de ce type : la relation qui décrit la structure de généralisation-spécialisation et la relation qui décrit la structure composé-composants."

"R.73. La relation de généralisation - spécialisation. Cette relation traduit la spécialisation d'une classe générale en d'autres classes qualifiées de spécialisées ou, à l'inverse, la généralisation de plusieurs classes spécialisées en une classe générale."

Signalons que seuls les attributs et les méthodes choisis par le concepteur (marqué d'un [H]) sont hérités de la classe plus générale aux classes plus spécifiques.

"R.80. La relation composé-composants traduit la représentation de la composition d'un objet simple composé, un objet complexe, une classe, une sous-couche ou une couche et elle est appliquée entre des objets, des classes, des couches, des sous-couches ou des combinaisons de ces éléments."

"R.82. A chaque extrémité d'une ligne de la structure de composé-composants, un intervalle de valeur (représentant les cardinalités) indique le nombre de composants que le composé peut avoir, et vice-versa.[...]."

"R.83. Si nécessaire, un carré ou un rectangle indiquant les contraintes topologiques obligatoires entre composé et composants est ajouté au dessous du triangle."

2.4.1.5.2. Les relations topologiques

Les relations topologiques sont un sous-ensemble des relations spatiales qui possèdent la propriété d'être préservées sous les transformations topologiques telles que rotation, translation, changement d'échelle. Nous reviendrons en détail sur ces relations importantes pour la modélisation de l'information géographique dans le chapitre 4.

"R.88. Le formalisme CONGOO définit deux relations topologiques de base : le voisinage (v) et la superposition (s). Elles peuvent être appliquées entre instances, objets, classes, couches et sous-couches, et leurs combinaisons."

"R.90. Les relations topologiques sont classifiées en deux catégories : en relations topologiques permises et en relations topologiques obligatoires. Les relations topologiques permises peuvent exister ou non entre les objets et les instances, tandis que les relations topologiques obligatoires doivent exister entre les objets. Ces dernières sont soit positives (obligation), soit négatives (interdiction)."

La relation de voisinage

En ce qui concerne la relation de voisinage des objets géo-graphiques simples :

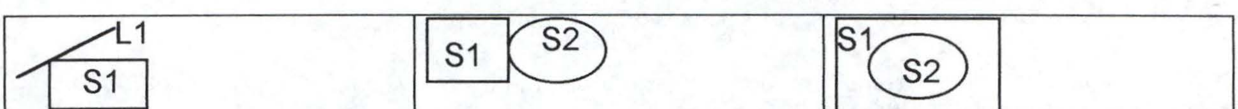
"R.100. La relation de voisinage nécessite la participation d'au moins une instance de type polygone(2D)."

"R.103. La relation de voisinage admet trois modalités : voisinage total (Vt), voisinage partiel (Vp) et voisinage nul (Vn). Cette dernière modalité trouve sa justification lors de la fixation de contraintes topologiques (voisinage interdit entre deux instances)"

"R.107. Le voisinage d'une instance 0D avec une instance 2D doit s'entendre comme la superposition de l'instance 0D sur la limite de l'instance 2D.[...]"

"R.108. Le voisinage d'une instance 1D avec une instance 2D doit s'entendre comme la superposition de l'instance 1D sur la limite de l'instance 2D.[...]"

"R.110. La relation de voisinage peut se manifester plusieurs fois entre les mêmes instances d'objets. Dans ce cas, les instances ne peuvent présenter qu'un voisinage partiel et l'on peut mentionner la fréquence de contacts entre les instances par un chiffre entre parenthèses (nombre de contact par un ou plusieurs points et/ou un ou plusieurs tronçons de lignes = suite de deux, ou plus, points contigus).[...]



voisinage partiel de S1 sur S2 et S2 sur S1 S1 Vp S2 et S2 Vp S1	voisinage partiel de S1 sur S2 et S2 sur S1 S1 Vp S2 et S2 Vp S1	voisinage nul de S1 sur S2 et S2 sur S1 S1 Vn S2 et S2 Vn S1

Figure 2.7 : Exemples de la relation topologique "voisinage"

En ce qui concerne la relation de voisinage des objets géo-graphiques composés et complexes de nouvelles règles apparaissent telles que voisinage partiel d'objets ponctuels non possibles pour des objets simples ponctuels.

La relation de superposition

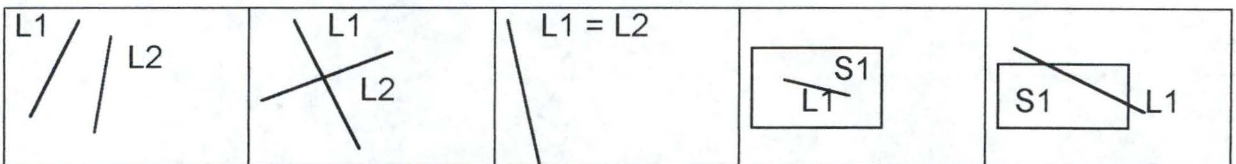
En ce qui concerne la relation de superposition des objets géo-graphiques simples :

"R.122. La relation de superposition, notée S, implique la cooccurrence (=coexistence partielle ou totale sur la même partie de l'espace) de deux ou plusieurs instances (ou "parties de ces instances). Sont exclues de la superposition, les situations ne faisant intervenir que la limite d'une instance 2D (traitées par la relation de voisinage)"

"R.124. La relation de superposition admet trois modalités : superposition totale (St),superposition partielle (Sp) et superposition nulle (Sn).[...]"

"R.125. La relation de superposition peut faire intervenir toutes les combinaisons types d'instances : point(0D), ligne(1D), polygone (2D), surface."

"R.133. La relation de superposition peut se manifester plusieurs fois entre les mêmes instances d'objets. Dans ce cas, les instances ne peuvent présenter qu'une superposition partielle et l'on peut mentionner et détailler la fréquence de cooccurrences (co) partielles entre les instances par un chiffre entre parenthèses accompagné des petits lettres "co". Si l'on dispose de l'information, on peut préciser le type de cooccurrences. Dans ce cas, les lettres (co) sont remplacées par le type : (p) pour points, (tro) pour tronçons, (zon) pour zones, etc.. Des combinaisons de ces trois abréviations sont possibles."



Superposition nulle de L1 sur L2 et L2 sur L1 L1 Sn L2 et L2 Sn L1	Superposittion partielle de L1 sur L2 et L2 sur L1 L1 Sp L2 et L2 Sp L1	Superposition totale de L1 sur L2 et L2 sur L1 L1 St L2 et L2 St L1	Superposition totale de L1 sur S1 et Superposition partielle de S1 sur L1 L1 St S1 S1 Sp L1	Superposition partielle de L1 sur S1 et de S1 sur L1 L1 Sp S1 S1 Sp L1

Figure 2.8 : Exemples de la relation topologique de superposition

En ce qui concerne la relation de superposition des objets géo-graphiques composés et complexes, de nouvelles règles apparaissent telles que superposition partielle d'objets ponctuels non possibles pour des objets simples ponctuels.

Des tableaux reprenant les relations topologiques possibles entre les différentes catégories des objets géo-graphiques simples (Figure 2.9), des objets géo-graphiques composés, des objets géo-graphiques complexes sont définis.

			
	+Sn,t -Sn,t - La relation "V" ne s'applique pas	+Sp +St -Sp -St +Sn -Sn - La relation "V" ne s'applique pas	+Vp +Vt -Vp -Vt +Sp +St -Sp -St +Vn -Vn +Sn -Sn -Voisinage = superpositon sur la limite du polygone
	+St +Sp -St -Sp +Sn -Sn - La relation "V" ne s'applique pas	+Sp,t +Sp,t -Sp,t -Sp,t +Sn -Sn - On peut préciser le type de superposition, par ex.: Sp(1p), Sp(max.0.5km),... - La relation "V" ne s'applique pas	+Vp,t +Vp,t -Vp,t -Vp,t +Sp +Sp,t -Sp -Sp,t +Vn -Vn +Sn -Sn - Le voisinage total dans ce cas concerne en principe des lignes fermées
	+Vt +Vp -Vt -Vp +St +Sp -St -Sp +Vn -Vn +Sn -Sn -Voisinage = superpositon sur la limite du polygone	+Vp,t +Vp,t -Vp,t -Vp,t +Sp,t +Sp -Sp,t -Sp +Vn -Vn +Sn -Sn - Le voisinage total dans ce cas concerne en principe des lignes fermées	+Sp,t +Sp,t -Sp,t -Sp,t +Vn,p -Vn,p +Sn -Sn

Figure 2.9 : Les relations topologiques entre OGS [Pantazis et Donnay, p 52]

En ce qui concerne les relations topologiques entre objets géo-graphiques / classes / couches / sous-couches :

"R.151. L'application des relations topologiques entre deux classes, couches, sous-couches, ou toute combinaison entre elles ou avec un ou plusieurs objets, indique que les classe(s), couche(s) ou sous-couche(s) sont considérées comme des OG dont les types dépendent des types et du nombre d'objets qu'elles contiennent. Les règles appliquées aux OGS, OGC et OGX sont également applicables à de tels types de relations, y compris la possibilité d'orienter l'application d'une relation."

2.4.1.5.3. Les relations logiques

"R.152. Les relations logiques concernent toutes les relations qui ne sont ni des relations topologiques, ni des relations de structure."

"R.154. Le formalisme CONGOO emprunte les règles du modèle E/R de MERISE, relatives à la signification, l'application et la notation des cardinalités dans le cadre des relations logiques."

Un tableau récapitulatif des relations du modèle CONGOO est proposées en annexe 1.

2.4.1.6. Les contraintes

Le modèle CONGOO emprunte les contraintes qui s'appliquent à une paire de relations au modèle E-A définies par [Nancy et al 90]. Ces contraintes d'intégrité sont l'exclusion, la simultanéité, la totalité, exclusion et totalité, inclusion et s'appliquent dans le modèle CONGOO aussi bien entre des relations de même catégorie (topologiques, logiques, de structure) qu'entre des relations de types différents.

Concernant les traitements, MECOSIG ne propose pas un modèle spécifique pour la modélisation des traitements et renvoie aux réseaux de Pétri, D.F.D.. Cependant les traitements à appliquer peuvent être intégrés à chaque objet, classe, couche, sous-couche et notés en-dessous des attributs des "objets".

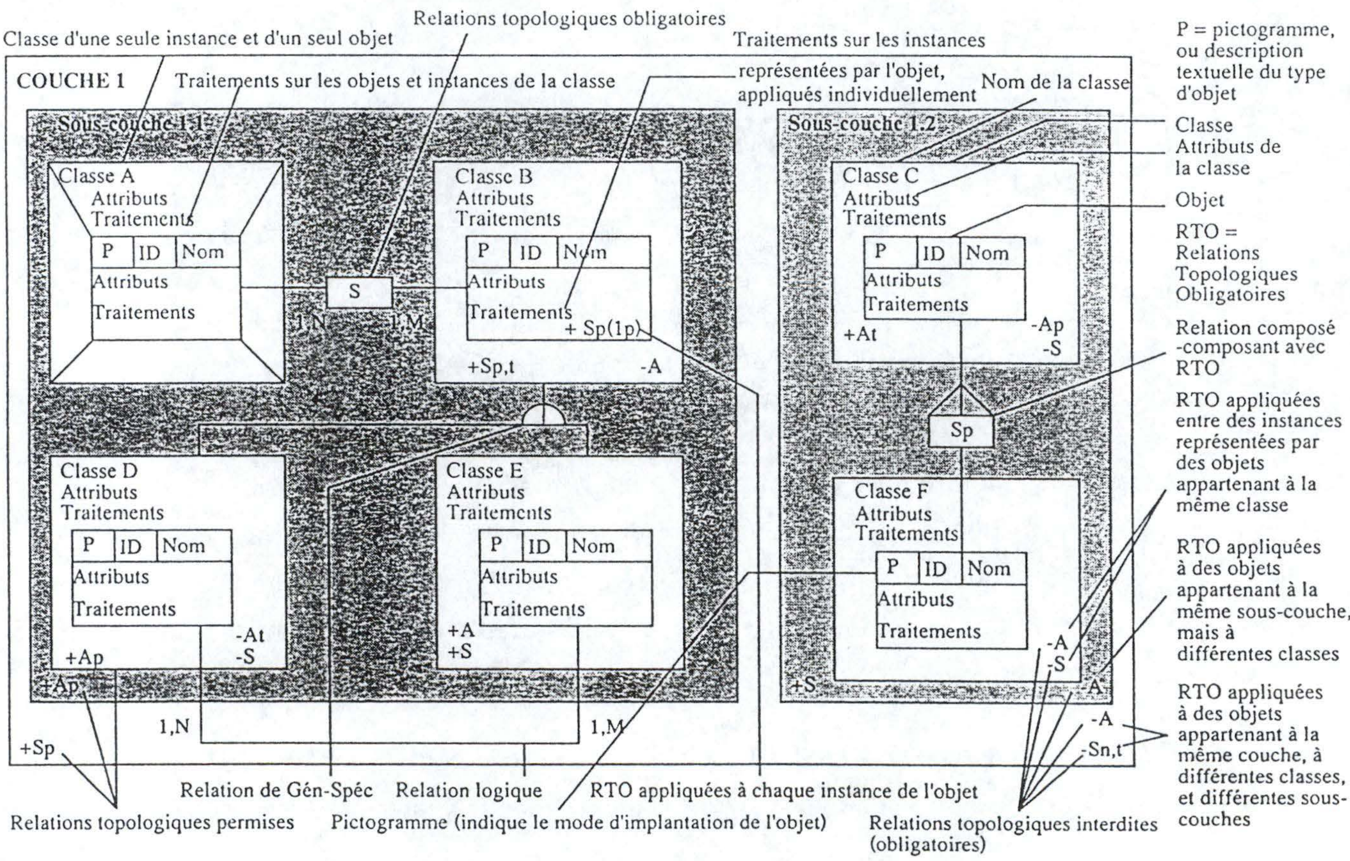


Figure 2.10 : Notation graphique des concepts de base de CONGOO [Pantazis et Donnay]

Ci-dessous, nous pouvons voir l'exemple vu plus haut, modélisé avec le modèle CONGOO d'après Pantazis.(Figure 2.11). Notons que pour des raisons de simplicité les attributs et les traitements (des objets et des classes d'objets) n'apparaissent pas. Une lecture sommaire de ce schéma est reprise en annexe 2.

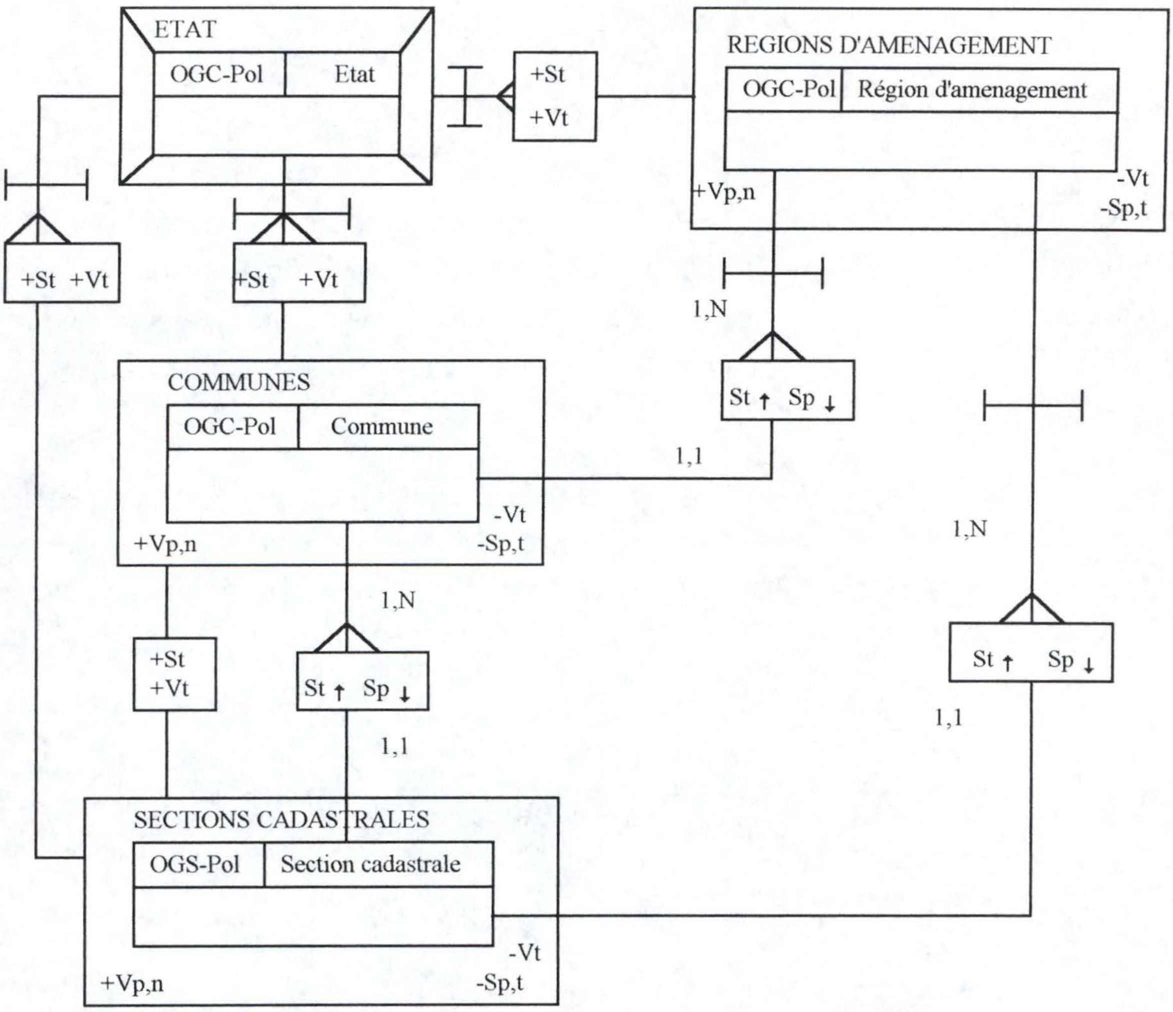


Figure 2.11 : Exemple de schéma effectué avec le modèle CONGOO [Pantazis et Donnay]

2.4.2. Critique du modèle CONGOO

Nous avons montré dans la partie 2.3.2 que le modèle E-A avait plusieurs lacunes pour modéliser l'information géographique. Différents modèles ont envisagé plusieurs solutions pour améliorer le modèle E-A. Nous pouvons remarquer ci-dessus que le modèle CONGOO a adopté la majorité de ces solutions et en a apporté d'autres notamment pour la représentation des relations topologiques.

Tout d'abord le modèle CONGOO prend en compte le mode d'implantation de l'objet par un pictogramme, la solution que nous avions évaluée comme la plus intéressante.

Ensuite nous avions vu qu'un bon modèle pour structurer l'information géographique devait posséder des mécanismes de structuration et une grande expressivité. Or le modèle

CONGOO propose divers mécanismes tels que l'héritage par la relation de généralisation/spécialisation mais aussi une forme d'agrégation par la relation de composé/composants. Il propose aussi les concepts de couche et de sous-couche qui permettent d'augmenter la lisibilité et une meilleure gestion de la complexité.

D'autre part l'utilisation de deux relations topologiques génériques bien définies et leurs représentations graphiques distinctes des représentations des autres relations, nous semblent fort intéressantes pour augmenter la compréhension et la lisibilité du schéma. Enfin la possibilité d'exprimer une relation entre objet et groupe d'objets (classe, couche, sous-couche) nous parait très utile pour la modélisation des données géo-graphiques.

Nous pouvons donc dire que ce modèle est très intéressant pour structurer l'information géographique vu sa grande expressivité, ses nombreux mécanismes de structuration et ses nouveaux apports tels que la représentation et la définition des relations topologiques, les relations entre groupe d'objets et les notions de sous-couche et couche.

Toutefois il nous semble qu'il y a un problème qui pourrait affecter l'efficacité de ce modèle. En effet ce modèle ne possède pas de sémantique formelle, or les définitions et les règles d'utilisation des concepts nous semblent parfois ambiguës et incomplètes. De plus certaines règles semblent parfois contradictoires. Tout ceci est très important car, comme nous l'avons déjà dit à diverses reprises, le modèle doit être à la base de la communication entre diverses personnes et est la fondation du S.I.G.. Un concept mal compris ou interprété différemment peut coûter très cher.

Nous pouvons regretter aussi la non représentation du temps souvent utile pour l'information géographique et l'absence d'un outil "CASE" pour ce modèle mais nous savons aussi que des recherches sont en cours sur ces domaines.

2.5. Conclusion

Nous avons montré dans ce chapitre que les méthodes de conception des S.I. "traditionnelles" ne fonctionnaient pas très bien pour la conception des S.I.G. vu leur spécificité et qu'il fallait donc des adaptations. Nous avons relevé les lacunes du modèle E-A pour la structuration de l'information géographique et ainsi mis en évidence un certain nombre de besoins pour un modèle de structuration de l'information géographique tels qu'une grande expressivité et la nécessité des mécanismes de structuration. Nous avons présenté le modèle CONGOO et montré qu'il semble bien convenir pour structurer l'information géographique mais qu'il n'avait pas de sémantique formelle. C'est pourquoi dans le chapitre suivant nous lui donnons une sémantique formelle.

Chapitre 3

Spécification formelle des concepts du Modèle CONGOO

3.1. Introduction

Nous avons montré que les langages de spécification devaient posséder différentes caractéristiques dont une interprétation formelle complète pour pouvoir jouer leurs rôles parfaitement. Dans le chapitre deux, nous avons vu que les concepts du modèle CONGOO étaient définis en langage naturel et que ces définitions ne semblaient pas toujours très claires. Il est donc important de définir une sémantique formelle du modèle CONGOO pour que ses concepts soient bien utilisés et que les schémas effectués avec le modèle CONGOO soient interprétés de manière univoque. Pour munir ce modèle d'une sémantique formelle, nous allons définir les concepts de ce modèle à l'aide d'un langage formel de spécification.

3.2. Présentation du langage formel de spécification utilisé

Le langage formel de spécification que nous utilisons est le langage GLIDER (General Language for the Incremental Definition an Elaboration of Requirements) [Dubois, Du Bois, Rifaut et Wodon 91]. GLIDER possède les trois principales caractéristiques demandées à un langage de spécification : un bon pouvoir d'expression, une mise à disposition de mécanismes puissants de structuration et des règles rigoureuses d'interprétation qui garantissent l'absence d'ambiguïté.

Une spécification GLIDER est organisée en ensemble de "clusters". Les clusters sont en général constitués d'une structure de données et des services offerts sur la structure de données. La structure de données est définie par une expression constituée de types de base prédéfinis (boolean, char, integer, string, real) et des types constructeurs prédéfinis (cartesien product, sequence, set, bag, union). Les services (opération) offerts sur la structure de données sont caractérisés au moyen de la logique du premier ordre. Signalons que des clusters peuvent aussi ne pas définir une structure de données, ils sont nommés opération clusters.

Deux mécanismes de structuration peuvent exister entre les clusters : l'héritage et la généricité. L'héritage dans GLIDER permet d'hériter l'expression de type, les invariants et les opérations d'un cluster à un autre. Cet héritage est simple et autorise la redéfinition d'opérations si elles préservent la sémantique originale. La généricité dans GLIDER permet de paramétrer les clusters. Un cluster paramétré est construit en utilisant un autre cluster appelé paramètre formel. Le mécanisme d'intanciation consiste à substituer au paramètre formel un paramètre actuel vérifiant les propriétés du paramètre formel.

GLIDER supporte la notion de spécification incomplète c'est-à-dire qu'une interprétation peut être donnée même à une spécification non terminée. GLIDER permet aussi d'exprimer des contraintes de temps réel (modélisation dynamique). Nous n'utiliserons pas l'aspect dynamique que nous offre GLIDER dans notre spécification.

Pour de plus amples renseignements nous vous renvoyons à [Dubois, Du Bois, Rifaut et Wodon 91] et [ICARUS 90].

3.3. Spécification formelle

Pour faciliter l'explication de la spécification, nous divisons notre présentation en deux parties. Dans la première partie nous ne considérons que les objets géo-graphiques simples et les objets non géo-graphiques des différentes catégories d'objets proposés par le modèle CONGOO. Par contre dans la deuxième partie nous essayons de tenir compte de toutes les catégories d'objets. Dans chacune des parties nous discutons la construction de la spécification formelle des concepts "majeures" (objet, classe, couche, sous-couche, relation logique et relation topologique) du modèle CONGOO et présentons des exemples de traduction des concepts dans le langage GLIDER. Les règles du modèle CONGOO étant très nombreuses (170 contenues dans 80 pages) nous n'en reprendrons qu'une partie pour illustrer nos propos. Signalons aussi que toutes les règles du modèle ne seront pas spécifiées formellement dans ce travail, seuls les concepts importants sont entièrement étudiés ici. Nous nous sommes fixé comme but de définir formellement principalement les concepts intervenant dans le schéma de la figure 3.1.

3.3.1. Formalisation des concepts de CONGOO en ne considérant que les d'objets géo-graphiques simples et les objets non géo-graphiques

3.3.1.1. Les Objets et leurs modes d'implantation

Comme nous l'avons déjà dit, le concept d'objet du modèle CONGOO peut être considéré comme un type dont la structure de données est composée d'un pictogramme, d'un identifiant, d'un certain nombre d'attributs logiques et graphiques auxquels sont (parfois) associés des traitements.

Nous pouvons construire un cluster objet dont l'expression de type est :

OBJET is CP[REPRESENTATION, IDENTIFIANT, AT_LOG, AT_GRAPH]

ou

AT_LOG is UNION[CP[], CP[Att1: TL], CP[Att1:TL, Att2:TL], CP[Att1:TL, Att2:TL, Att3:TL],...,CP[Att1:TL, Att2:TL,...,Attn:TL]]

TL is UNION[BOOLEAN, INTEGER, CHAR, STRING, REAL,...]

AT_GRAPH is UNION[CP[], CP[Att1: TG], CP[Att1:TG, Att2:TG], CP[Att1:TG, Att2:TG, Att3:TG],...,CP[Att1:TG, Att2:TG,...,Attn:TG]]

TG is UNION[COULEUR, EPAISSEURTRAIT,...]

COULEUR is ENUM('rouge', 'bleu', 'vert', 'jaune')

EPAISSEURTRAIT is ('1pt', '2pt', '3pt', '4pt', '5pt', '6pt', '7pt', '8pt')

Pour spécifier les pictogrammes, nous avons dû rechercher leurs significations exactes. Ils signifient suivant l'élément ou les éléments géométriques représentés par le pictogramme que l'objet a une représentation de type point, ligne, polygone ou une composition ou combinaison de ces éléments. Ainsi, un pictogramme où est dessiné un point indique que toute occurrence de l'objet sera représentée par un élément géométrique de type point. Nous allons donc spécifier un nouveau cluster appelé représentation pour définir les différents pictogrammes :

REPRESENTATION is UNION[POINT, LIGNE, POLYGONE, ALPHA,...]

ALPHA étant la notation proposée quand il n'y a pas de représentation graphique c'est-à-dire quand c'est le pictogramme d'un objet non-géographique.

Les notions de point, ligne, polygone, surface ne sont pas définies dans le modèle CONGOO mis à part le dessin repris ci-dessous et la règle 7.

Point	Ligne	Polygone	Surface
(0 Dimension)	(1 Dimension)	(2 Dimensions)	("espace continu"), 2D
Représentation par	exemple de :		
Ville	Rivière	Forêt	Surface de tendance

"R.7. Un OGS du type polygone (OGS-Pol) est délimité par une ligne fermée. Une ligne fermée est la ligne dont le début et la fin coïncident en un même point. La ligne qui définit le périmètre du polygone fait partie du polygone, et elle n'est plus considérée comme un OGS-L."

Le modèle utilise les caractéristiques d'un point, d'une ligne, d'un polygone à plusieurs reprises. Par exemple, la phrase dans la règle 7, "une ligne fermée est la ligne dont le début et la fin coïncident en un même point suppose donc que la ligne a un début. Plusieurs questions peuvent être posées sur les caractéristiques de ces éléments géométriques sans trouver de réponse explicite : Une ligne se coupant est-elle considérée comme une ligne ? Une ligne fermée peut-elle se couper ? Une ligne est-elle constituée d'une suite de segments ? Un polygone peut-il avoir des "trous" ?

De plus les caractéristiques de ces éléments géométriques sont utilisées pour savoir si une relation topologique spécifique peut exister entre tel ou tel élément géométrique (exemple : pas de superposition partielle d'un point sur une ligne ou pas de voisinage d'un point avec une ligne...) .

Pour toutes ces raisons, il nous paraît important pour notre spécification formelle de définir les caractéristiques (minimun) des différents éléments géométriques.

A propos de la représentation géométrique choisie pour un objet, il nous semble important de bien faire la distinction entre la modélisation de l'objet réel et le choix de leur représentation géométrique qui rappelons-le n'a souvent rien à voir avec sa forme réelle. Il est important de bien comprendre que, par exemple, un "objet de type point" est un objet "réel" pour lequel nous avons choisi une représentation géométrique de type point. Ainsi un bâtiment peut être considéré comme un "objet de type point" si on a décidé de le représenter par un point et cet objet bâtiment peut très bien avoir un attribut surperficie. A notre avis, il serait donc préférable pour qu'il n'y ait pas de confusion de dire par exemple dans la règle 7 que la représentation d'un objet du type polygone est délimitée par une ligne fermée ou encore qu'un objet du type polygone a une représentation délimitée par une ligne fermée au lieu de dire "Un objet du type polygone est délimité par une ligne fermée".

Nous avons donc décidé d'essayer de préciser les caractéristiques qu'il faut donner à un point, une ligne et un polygone (les modes d'implantations composés et complexes seront étudiés dans la partie 3.3.2). Le choix du minimum de ces caractéristiques est très difficile à effectuer et a déjà été posé à plusieurs reprises : "What are the fondamental geometric properties of geographic objects needed to describe their relations?" [Abler 87]

Pour faire ces choix et respecter l'idée de l'auteur du modèle nous avons recherché tous les éléments de réponse qui existent dans le modèle et les exemples d'utilisation de celui-ci.

Nous avons défini les caractéristiques des éléments géométriques pour un espace à deux dimensions en ayant une vision topologique de l'espace c'est-à-dire que les relations qualitatives spatiales entre les objets sont considérées comme plus importantes que leurs formes précises et leurs positions absolues. Il nous a donc paru peu intéressant de définir le type point par ces coordonnées.

Soulignons que la définition d'un point est un problème philosophique et mathématique complexe que nous n'avons pas essayé de résoudre ici.

Nous définissons le cluster point comme ceci :

CLUSTER POINT
exports all
is
asserts
un point ne peut avoir de partie
$\forall z\ [C(z, x) \Rightarrow C(z, y)] \Rightarrow Egal(x, y)$
hints
z, x, y : POINT

OPERATION Egal *égalité entre deux points*
arity POINT $\times$ POINT $\rightarrow$ BOOLEAN
asserts
Egal (x, x)
Egal (x, y) $\Rightarrow$ Egal (y, x)
Egal (x, y) $\wedge$ Egal (y, z) $\Rightarrow$ Egal (x, z)
hints
x, y, z : POINT
OPERATION Sup *superposition entre deux points*
arity POINT $\times$ POINT $\rightarrow$ BOOLEAN
pré
post Sup(x, y) $\equiv$ Egal(x, y)
hints
x, y : POINT
OPERATION C *connecté c'est-à-dire partage un point en commun ou en contact*
arity POINT $\times$ POINT $\rightarrow$ BOOLEAN
asserts

```
 C(x, x)
 C(x, y) ⇒ C(y, x)
hints
 x,y : POINT
OPERATION Ce *contact externe entre deux points*
arity POINT × POINT → BOOLEAN
asserts
 Ce(x, y) ≡ C(x, y) ∧ ¬ (Egal (x, y))
hints
 x, y : POINT
OPERATION Distance *Distance entre deux points*
arity POINT × POINT → REAL
asserts
 Distance (x, y) ≥ 0
 Distance(x, y) = Distance(y, x)
 Egal(x, y) ⇔ Distance(x, y) = 0
 ∀z Distance(x, y) ≤ Distance(x, z) + Distance(z, y)
hints
 x, y, z : POINT
OPERATION ∈ *est "membre" de*
arity POINT × POINT → BOOLEAN
pré
post x∈ y ≡ Egal(x, y)
hints
 x, y : POINT
END POINT
```

Le cluster ligne est défini comme une séquence de deux points au moins où chaque point de la séquence est connecté extérieurement avec le point précédent et le suivant. Vu que la ligne peut se croiser, nous n'interdirons pas que deux points identiques appartiennent à la séquence s'ils ne sont pas voisins dans la séquence. Le cas où tous les points en contact externe avec un point d'une ligne appartiendraient à cette ligne n'est pas possible.

```
CLUSTER LIGNE
 exports all
 is SEQ[POINT]
asserts
*une ligne doit être composée d'au moins deux points*
Length(l) ≥ 2
*tout point d'une ligne est en contact externe avec le point précédent et le point suivant de la
ligne*
∀ i : 1< i < Length(l) ⇒ Ce[Si(l, i),Si(l, i+1)] ∧ Ce[Si(l, i-1), Si(l, i)] ∧ Ce[Si(l, 1),Si(l, 2)] ∧
Ce[Si(l, Length(1)), Si(l, Length(l)-1] *'Si' est une opération d'accès dans une séquence*
*tous les points en contact externe à un point d'une ligne ne peuvent pas tous appartenir à cette
ligne*
∀ x, y : x ∈ l ∧ Ce(x, y) ⇒ ∃ y : y ∉ l
hints
x,y : POINT
```

l : LIGNE
i : INTEGER

OPERATION Fermée? *si une ligne est fermée ou pas*
arity LIGNE $\rightarrow$ BOOLEAN
pré
post Fermée?(l) $\equiv$ Sup(First(l), Last(l))
hints
l : LIGNE
OPERATION = *égalité entre deux lignes*
arity LIGNE $\times$ LIGNE $\rightarrow$ BOOLEAN
pré
post l = l' $\Leftrightarrow$ (Length(l) = Length(l') $\wedge$ ($\forall$i : 1 $\leq$ i $\leq$ Length(l) $\Rightarrow$ Egal(Si(l, i), Si(l', i)) $\vee$
$\forall$i : 1 $\leq$ i $\leq$ Length(l) $\Rightarrow$ Egal(Si(l, i), Si(l', (Length(l')-i + 1)))))
hints
l,l' : LIGNE
i : INTEGER
OPERATION Finsligne *fonction qui donne les deux points limites d'une ligne si elle est non fermée*
arity LIGNE $\rightarrow$ CP(POINT, POINT)
pré $\neg$(Fermée?(l))
post Finligne(l) = <Last(l), First(l)>
hints
l : LIGNE
END LIGNE

Dans le modèle, il est précisé qu'un objet géo-graphique de type ligne peut être orienté. Nous avons donc créé un cluster ligne orienté qui est différent du cluster ligne par trois opérations : la première est l'opération d'égalité, elle est considérée comme vraie entre deux lignes si les séquences des points les formant sont identiques ou inverses; par contre pour les lignes orientées, elle est vraie seulement si les séquences de points sont identiques. Les deux autres opérations permettent pour une ligne orientée de trouver le début ou la fin de la ligne, ce qu'il n'est pas possible de déterminer pour une ligne non orientée.

CLUSTER LIGNEORIENTE
exports all
is LIGNE
only - gets Fermée?
asserts
hints

OPERATION = *égalité entre deux lignes orientées*
arity LIGNEORIENTE $\times$ LIGNEORIENTE $\rightarrow$ BOOLEAN
pré
post l = l' $\Leftrightarrow$ (Length(l) = Length(l') $\wedge$ ($\forall$i : 1 $\leq$ i $\leq$ Length(l) $\Rightarrow$ Egal(Si(l, i), Si(l', i)))
hints
l,l' : LIGNEORIENTE
i : INTEGER

OPERATION Débutligne *fonction qui donne le début de la ligne*
arity LIGNEORIENTE $\rightarrow$ POINT
pré $\neg$(Fermée?(l))
post Débutligne(l) = First(l)
hints
l : LIGNEORIENTE
OPERATION Finligne *fonction qui donne la fin de la ligne*
arity LIGNEORIENTE $\rightarrow$ POINT
pré $\neg$(Fermée?(l))
post Finligne(l) = Last(l)
hints
l : LIGNEORIENTE
END LIGNEORIENTE

En ce qui concerne la représentation de type polygone, il important de souligner qu'ils ne peuvent pas avoir de trous. Nous la définissons comme :

CLUSTER POLYGONE
exports all
is CP[Intérieur : SET[POINT], Frontière : LIGNE]
asserts
l'intérieur et la frontière d'un polygone ne peuvent être vides
$\neg$(Empty?(Intérieur(pol)) $\wedge$ $\neg$(Empty?(Frontière(pol))
tous les points connectés à un point de l'intérieur d'un polygone appartiennent à ce polygone
$\forall$ x, y : x $\in$ Intérieur(pol) $\wedge$ C(x, y) $\Rightarrow$ y $\in$ Intérieur(pol) $\vee$ y $\in$ Frontière(pol)
tout point de la frontière d'un polygone est connecté extérieurement avec au moins un point de l'intérieur de ce polygone et au moins un point n'appartenant pas à ce polygone
$\forall$ x : x $\in$ Frontière(pol) $\Rightarrow$ $\exists$ y : y $\in$ Intérieur(pol) $\wedge$ Ce(x, y) $\wedge$ $\exists$ z : $\neg$(z $\in$ pol) $\wedge$ Ce(x, z) $\wedge$ z $\neq$ y

la limite d'un polygone est une ligne fermée
Egal[Si(Frontière(pol),1), Si(Frontière(pol), Length(Frontière(pol)))]
* la limite est une ligne qui ne peut être croisée*
$\forall$i, j : 1< i, j <Length[Frontière(pol)] $\wedge$ i $\neq$ j
$\Rightarrow$ $\neg$[Egal[Si(Frontière(pol),i), Si(Frontière(pol),j)]]
hints
x, y, z : POINT
i, j : INTEGER
pol : POLYGONE

OPERATION $\in$ *est membre de*
arity POINT $\times$ POLYGONE $\rightarrow$ BOOLEAN
pré
post y $\in$ pol $\Leftrightarrow$ y $\in$ Intérieur(pol) $\vee$ y $\in$ Frontière(pol)
hints
y : POINT
pol : POLYGONE
OPERATION = *égalité entre deux polygones*
arity POLYGONE $\times$ POLYGONE $\rightarrow$ BOOLEAN
pré

post pol = pol' $\Leftrightarrow$ Intérieur(pol) = Intérieur(pol') $\wedge$ Frontière(pol) = Frontière(pol')
hints
pol, pol' : POLYGONE
END POLYGONE

Nous pouvons à présent spécifier le cluster représentation dont l'expression de type est :
REPRESENTATION
is UNION[POINT, LIGNE, LIGNEORIENTE, POLYGONE, ALPHA,...]

Dans ce cluster nous définissons les "opérations" topologiques de voisinage et de superposition. Le modèle définit les contraintes topologiques entre les instances d'objets mais vu que nous distinguons pour la spécification formelle la représentation géométrique de l'objet et l'objet, il nous paraît normal de définir les contraintes topologiques au niveau de la représentation. Nous les définirons aussi au niveau de l'objet car elles sont utilisées à ce niveau dans le modèle.

Etudions donc les règles concernant les différentes relations topologiques du modèle. Nous avons vu que deux relations principales sont définies : la relation de voisinage et la relation de superposition. Reprenons les propos de l'auteur à propos de la définition de la relation de voisinage (nous vous renvoyons aussi aux règles énoncé chapitre 2):

"R.99. Selon le type d'instances mises en présence, la relation de voisinage, notée V, peut s'interpréter en termes d'adjacence, de partage de limite ou de superposition sur une limite. Pour éviter toute confusion, le formalisme ne retient la propriété de voisinage que lorsqu'une instance de type polygone (2D), au moins, participe à la relation. Plus précisément encore, c'est la superposition de la limite de cette instance (0D ou 1D) ou limite d'instance (2D) qui définit la relation de voisinage."

Si nous définissons l'opération comme une vérification de la relation de voisinage entre deux instances du type représentation nous pouvons définir l' "arity" de l'opération par :

REPRESENTATION$\times$REPRESENTATION $\rightarrow$ BOOLEAN

Relevons que la relation de voisinage ne peut exister que si les instances du type représentation ont une représentation graphique et qu'au moins une des deux instances soit de type polygone. Ceci sera inscrit dans la précondition d'utilisation de cette opération. La relation de voisinage existe entre deux instances de type polygone s'il existe au moins un point de la limite d'une des instances polygones superposé à un point de la limite de l'autre instance. La superposition entre deux points a été définie dans le cluster point comme étant équivalente à l'égalité entre ces deux points. Le voisinage entre deux polygones (deux instances de type polygone) peut donc être défini formellement par :

V(pol, pol') $\equiv \exists$ x, y : x $\in$ Frontière(pol) $\wedge$ y $\in$ Frontière(pol') $\wedge$ Sup(x, y)
où pol et pol' sont de type polygone et x et y sont de type point

La relation de voisinage entre une ligne (ou point) et un polygone est réalisée si au moins un point de la ligne (ou point) est superposé à un point de la limite du polygone.

Trois modalités sont distinguées pour la relation de voisinage : voisinage total (Vt), voisinage partiel (Vp) et voisinage nul (Vn). Remarquons que ces notions, il est vrai relativement intuitives, ne sont pas définies dans le modèle. Il est indiqué seulement que les trois modalités ne sont pas applicables entre tous les types de représentation d'objets (chapitre 2 figure 2.9). La relation de voisinage total peut être réalisée entre deux polygones si chaque point de la limite d'un polygone est superposé à un point de la limite de l'autre. Elle est réalisée entre une ligne (ou point) et un polygone si chaque point de la ligne (ou le point) est superposé à un point de la limite du polygone et l'inverse pour le voisinage total entre un polygone et une ligne. La relation de voisinage partiel est réalisée si la relation de voisinage est réalisée mais pas la relation de voisinage total. La relation de voisinage nul est réalisée quand la relation de voisinage n'est pas vraie. Remarquons que l'on peut déduire plusieurs propriétés de ces définitions telles que les relations de voisinage total et de voisinage partiel ne sont pas symétriques, qu'un point ne peut être voisin partiellement à un polygone ou une ligne,...Ces propriétés seront étudiées en détail dans le chapitre quatre.

La relation de superposition est définie par :

"R.122. La relation de superposition, notée S, implique la cooccurence (=coexistence partielle ou totale sur la même partie de l'espace) de deux ou plusieurs instances (ou "parties de ces instances). Sont exclues de la superposition, les situations ne faisant intervenir que la limite d'une instance 2D (traitées par la relation de voisinage)"

L'opération de superposition a la même "arity" que les opérations de voisinages. La relation de superposition entre deux polygones est réalisée s'il existe un point d'un polygone superposé à un point de l'autre et que ces deux points n'appartiennent pas tous les deux à la limite du polygone (car il s'agirait alors de la relation de voisinage). La relation est vraie entre une ligne (ou un point) et un polygone s'il existe un point de la ligne (ou le point) superposé à un point du polygone mais pas sur sa limite et elle est vraie entre un point et une ligne si le point est superposé à un point de la ligne.

Trois modalités sont aussi distinguées pour la propriété de superposition : superposition totale (St), superposition partielle (Sp) et superposition nulle (Sn). Ces trois modalités ne sont à nouveau pas définies. Or cette fois, elles nous semblent moins intuitives car d'aprés les exemples que nous avons étudiés il semble par exemple qu'une ligne peut être superposée totalement à un polygone en étant voisine partiellement de ce polygone; par contre elle ne peut être voisine totalement et superposée à un polygone. La superposition totale entre une ligne et un polygone semble donc être réalisée (ceci devrait être confirmé par l'auteur du modèle) si tous les points de la ligne sont superposés à un point du polygone et qu'au moins un point de la ligne soit superposé à un point qui n'appartient pas à la limite du polygone. La superposition totale d'un polygone avec un autre polygone implique que tous les points du premier polygone soient superposés à des points du second. Un point est superposé totalement à un polygone si le point est superposé à un point du polygone n'appartenant pas à la limite de celui -ci. Signalons aussi que les relations de superposition partielle et totale ne sont pas symétriques.

L'auteur donne aussi comme relation topologique la négation des relations que nous venons de voir c'est-à-dire la "non" superposition totale, le "non" voisinage total, ...

Nous pouvons voir dans le tableau de l'auteur Figure 2.9 que toutes les relations topologiques ne peuvent pas être appliquées entre toute les combinaisons d'éléments géométriques. Par exemple un polygone ou une ligne ne peut être superposé totalement à un point de même un point ne peut pas être superposé partiellement à un polygone etc... Nous pouvons regretter seulement que dès qu' une relation topologique ne peut s'appliquer entre deux éléments géométriques, sa relation négative ne puisse pas s'appliquer. Par exemple la relation de superposition partielle entre un point et une ligne ne peut exister. Du coup, pour l'auteur, la non superposition partielle entre un point et une ligne ne peut être utilisée bien que cette relation puisse exister. Dans la suite nous suivrons l'idée de l'auteur du modèle même si nous ne sommes pas tout à fait d'accord sur ce point.

Définissons à présent le cluster représentation.

CLUSTER REPRESENTATION *représentation des objets*
exports all
is UNION[POINT, LIGNE, LIGNEORIENTE, POLYGONE, ALPHA,...]
asserts
hints

- -

pour ne pas surcharger les définitions des opérations nous n'avons pas tenu compte explicitement du type LIGNEORIENTE mais les définitions sont les mêmes que pour le type LIGNE
OPERATION S(r, r') *superposition entre deux représentations graphiques*
arity REPRESENTATION $\times$ REPRESENTATION $\rightarrow$ BOOLEAN
pré $\neg$(Is-of-ALPHA(r)) $\wedge$ $\neg$(Is-of-ALPHA(r'))
post
(Is-of-POINT(r) $\vee$ Is-of-LIGNE(r))$\wedge$ (Is-of-POINT(r') $\vee$ Is-of-LIGNE(r'))
$\Rightarrow$ S(r, r') $\equiv \exists$ x,y : x $\in$ r $\wedge$ y $\in$ r' $\wedge$ Sup(x, y)

(Is-of-POINT(r) $\vee$ Is-of-LIGNE(r)) $\wedge$ (Is-of-POLYGONE(r'))
$\Rightarrow$ S(r, r') $\equiv \exists$ x,y : x $\in$ r $\wedge$ y $\in$ Intérieur(r') $\wedge$ Sup(x, y)

(Is-of-POLYGONE(r)) $\wedge$ (Is-of-POINT(r') $\vee$ Is-of-LIGNE(r'))
$\Rightarrow$ S(r, r') $\equiv \exists$ x,y : x $\in$ Interieur(r) $\wedge$ y $\in$ r' $\wedge$ Sup(x, y)

(Is-of-POLYGONE(r)) $\wedge$ (Is-of-POLYGONE(r'))
$\Rightarrow$ S(r, r') $\equiv \exists$ x,y : x $\in$ r $\wedge$ y $\in$ r' $\wedge$ Sup(x, y) $\wedge$ $\neg$ (x $\in$ Frontière(r) $\wedge$ y $\in$ Frontière(r'))

hints

r, r' : REPRESENTATION

x,y : POINT

OPERATION +Sn(r, r') *superposition nulle entre deux représentations graphiques*

arity REPRESENTATION $\times$ REPRESENTATION $\rightarrow$ BOOLEAN

pré $\neg(\text{Is-of-ALPHA}(r)) \wedge \neg(\text{Is-of-ALPHA}(r'))$

post $+\text{Sn}(r, r') \equiv \neg(\text{S}(r, r'))$

hints

r, r': REPRESENTATION

OPERATION +St(r, r') *superposition totale entre deux représentations graphiques*

arity REPRESENTATION $\times$ REPRESENTATION $\rightarrow$ BOOLEAN

pré $\neg(\text{Is-of-ALPHA}(r)) \wedge \neg(\text{Is-of-ALPHA}(r')) \wedge$

$\neg((\text{Is-of-LIGNE}(r) \vee \text{Is-of-POLYGONE}(r)) \wedge \text{Is-of-POINT}(r')) \wedge$

$\neg((\text{Is-of-POLYGONE}(r)) \wedge (\text{Is-of-LIGNE}(r'))$

post

$(\text{Is-of-POINT}(r) \vee \text{Is-of-LIGNE}(r)) \wedge (\text{Is-of-LIGNE}(r'))$

$\Rightarrow +\text{St}(r, r') \equiv \forall\, x\, \exists\, y : x \in r \wedge y \in r' \wedge \text{Sup}(x, y)$

$(\text{Is-of-POINT}(r) \vee \text{Is-of-LIGNE}(r)) \wedge (\text{Is-of-POLYGONE}(r'))$

$\Rightarrow +\text{St}(r, r') \equiv \forall\, x\, \exists\, y : x \in r \wedge y \in r' \wedge \text{Sup}(x, y) \wedge \text{S}(r, r')$

$(\text{Is-of-POINT}(r) \wedge \text{Is-of-POINT}(r') \Rightarrow \text{Sup}(r,r'))$

$(\text{Is-of-POLYGONE}(r)) \wedge (\text{Is-of-POLYGONE}(r'))$

$\Rightarrow \text{S}(r, r') \equiv \forall\, x\, \exists\, y: x \in r \wedge y \in r' \wedge \text{Sup}(x, y)$

hints

r, r' : REPRESENTATION

x,y : POINT

OPERATION +Sp(r, r') *superposition partielle entre deux représentations graphiques*

arity REPRESENTATION $\times$ REPRESENTATION $\rightarrow$ BOOLEAN

pré $\neg(\text{Is-of-ALPHA}(r)) \wedge \neg(\text{Is-of-ALPHA}(r')) \wedge \neg\, (\text{Is-of-POINT}(r))$

post

$((\text{Is-of-LIGNE}(r) \vee \text{Is-of-POLYGONE}(r)) \wedge \text{Is-of-POINT}(r'))$

$\vee\, (\text{Is-of-POLYGONE}(r) \wedge \text{Is-of-LIGNE}(r'))$

$\Rightarrow +\text{Sp}(r, r') \equiv \text{S}(r, r')$

$(\text{Is-of-LIGNE}(r) \wedge \text{Is-of-LIGNE}(r')) \vee ((\text{Is-of-POLYGONE}(r) \vee \text{Is-of-LIGNE}(r)) \wedge \text{Is-of-POLYGONE}(r')) \Rightarrow +\text{Sp}(r, r') \equiv \text{S}(r, r') \wedge \neg(+\text{St}(r, r'))$

hints

r, r' : REPRESENTATION

OPERATION V(r, r') *voisinage entre deux représentations graphiques*

arity REPRESENTATION $\times$ REPRESENTATION $\rightarrow$ BOOLEAN

pré $\neg(\text{Is-of-ALPHA}(r)) \wedge \neg(\text{Is-of-ALPHA}(r')) \wedge$

$\neg((\text{Is-of-POINT}(r) \vee \text{Is-of-LIGNE}(r)) \wedge (\text{Is-of-POINT}(r') \vee \text{Is-of-LIGNE}(r'))$

post

$((\text{Is-of-POINT}(r) \vee \text{Is-of-LIGNE}(r)) \wedge \text{Is-of-POLYGONE}(r'))$

$\Rightarrow \text{V}(r, r') \equiv \exists x, y : x \in r \wedge y \in \text{Frontière}(r') \wedge \text{Sup}(x, y)$

$\equiv \text{S}(r, \text{Frontière}(r'))$

(Is-of-POLYGONE(r)∧ (Is-of-POINT(r') ∨ Is-of-LIGNE(r')))
⇒ V(r, r') ≡ S(Frontière(r), r')

(Is-of-POLYGONE(r) ∧ Is-of-POLYGONE(r'))
⇒ V(r, r') ≡ S(Frontière(r), Frontière(r'))
hints
x, y : POINT
r, r' : REPRESENTATION
OPERATION +Vt(r, r') *voisinage total entre deux représentations graphiques*
arity REPRESENTATION × REPRESENTATION → BOOLEAN
pré ¬(Is-of-ALPHA(r)) ∧ ¬(Is-of-ALPHA(r')) ∧
¬((Is-of-POINT(r) ∨ Is-of-LIGNE(r))∧ (Is-of-LIGNE(r'))) ∧
¬(Is-of-POINT(r'))
post
((Is-of-POINT(r) ∨ Is-of-LIGNE(r)) ∧ Is-of-POLYGONE(r'))
⇒ +Vt(r, r') ≡ +St(r, Frontière(r'))

(Is-of-POLYGONE(r) ∧ Is-of-LIGNE(r'))
⇒ +Vt(r, r') ≡ +St(Frontière(r), r')

(Is-of-POLYGONE(r)∧Is-of-POLYGONE(r'))
⇒ +Vt(r, r') ≡ +St(Frontière(r), Frontière(r'))
hints
r, r' : REPRESENTATION
OPERATION +Vn(r, r') *voisinage nul entre deux représentations graphiques*
arity REPRESENTATION × REPRESENTATION → BOOLEAN
pré ¬(Is-of-ALPHA(r)) ∧ ¬(Is-of-ALPHA(r')) ∧
¬((Is-of-POINT(r) ∨ Is-of-LIGNE(r))∧ (Is-of-POINT(r') ∨ Is-of-LIGNE(r')))
post +Vn(r, r') ≡ ¬(V(r, r'))
hints
r, r' : REPRESENTATION
OPERATION +Vp(r, r') *voisinage partiel entre deux représentations graphiques*
arity REPRESENTATION × REPRESENTATION → BOOLEAN
pré ¬(Is-of-ALPHA(r)) ∨ ¬(Is-of-ALPHA(r')) ∧ ¬(Is-of-POINT(r)) ∧
¬(Is-of-LIGNE(r) ∧ (Is-of-POINT(r') ∨ Is-of-LIGNE(r')))
post
(Is-of-POLYGONE(r) ∧ (Is-of-POINT(r'))
⇒ +Vp(r, r') ≡ V(r, r')

(Is-of-LIGNE(r) ∧ Is-of-POLYGONE(r')) ∨ (Is-of-POLYGONE(r)∧ Is-of-LIGNE(r')) ∨
(Is-of-POLYGONE(r)∧Is-of-POLYGONE(r'))
⇒ +Vp(r, r') ≡ V(r, r') ∧ ¬(+Vt(r, r'))
hints
r, r' : REPRESENTATION
OPERATION Dim *donne la dimension d'une représentation*
arity REPRESENTATION → INTEGER
pré ¬(Is-of-ALPHA(r))
post

```
Dim(r) = i ∧
(Is-of-POINT(r) ⇒ i = 0 ∨
(Is-of-LIGNE(r) ∨ Is-of-LIGNEORIENTE(r)) ⇒ i = 1 ∨ Is-of-POLYGONE(r) ⇒ i = 2)
 hints
r : REPRESENTATION
i : INTEGER
END REPRESENTATION
```

Les différents clusters que nous venons de réaliser nous permettent de définir les objets avec une représentation simple.

```
CLUSTER OBJET
 exports all
is  CP[Représentation : REPRESENTATION, Surrogate : IDENTIFIANT, Alogique :
AT_LOG, Agraphique : AT_GRAPH]
 asserts
  ∀ o, o' : o ≠ o' ⇒ Surrogate(o) ≠ Surrogate(o')
 hints
  o, o' : OBJET
-----------------------------------------------------------------
END OBJET
```

Des opérations classiques de création et de modification doivent normalement être définies dans la plupart des clusters que nous allons voir mais nous ne l'avons pas fait ici par manque de place.

```
CLUSTER ONONG *objet non géo-graphique c'est-à-dire sans représentation graphique*
 exports all
 is OBJET
 asserts
Is-of-ALPHA[Représentation(o)] ∧ Agraphique(o) = CP[]
 hints
o : ONONG
-----------------------------------------------------------------
END ONONG
```

```
CLUSTER OGS_P *objet géo-graphique simple du type point*
 exports all
 is OBJET
 asserts
*un objet géo-graphique simple de type point a une représentation de type point*
Is-of-POINT[Représentation(o)]
 hints
  o : OGS_P
-----------------------------------------------------------------
END OGS_P
```

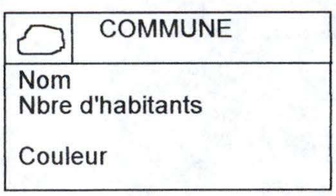
CLUSTER OGS_L *objet géo-graphique simple du type ligne*
exports all
is OBJET
asserts
un objet géo-graphique simple de type ligne a une représentation de type ligne ou de type ligne orientée
Is-of-LIGNE[Représentation(o)] $\vee$ Is-of-LIGNEORIENTE[Représentation(o)]
hints
o : OGS_L

END OGS_L

CLUSTER OGS_POL *objet géo-graphique simple du type polygone*
exports all
is OBJET
asserts
un objet géo-graphique simple de type polygone a une représentation de type polygone
Is-of-POLYGONE[Représentation(o)]
hints
o : OGS_POL

END OGS_POL

Exemple de traduction d'une notation à une autre pour un objet :



COMMUNE is OGS_POL
asserts
$\forall$c : Is-of-(CP[Nom : STRING, Nbre d'habitants : INTEGER])[Alogique(c)]
$\forall$c : Is-of-(CP[Couleur : COULEUR])[Agraphique(c)]
hints c : COMMUNE

Nous allons définir un cluster OGS (objet géo-graphique simple) reprenant les objets géo-graphiques de type point, de type ligne et de type polygone dans lequel on va définir les opérations topologiques pour les objets.

CLUSTER OGS *objet géo-graphique simple*
exports all
is UNION[OGS_P, OGS_L, OGS_POL]
asserts
hints

OPERATION +S(o, o') *superposition entre deux objets géo-graphiques simples*
arity OGS $\times$ OGS $\rightarrow$ BOOLEAN
pré
post +S(o, o') $\Leftrightarrow$ S(Représentation(o), Représentation(o'))
hints
o, o' : OGS

OPERATION -S(o, o') *non superposition entre deux objets géo-graphiques simples*
arity OGS × OGS → BOOLEAN
pré
post -S(o, o') ⇔ ¬(+S(o, o'))
hints
o, o' : OGS
OPERATION +Sn(o, o') *superposition nulle entre deux objets géo-graphiques simples*
arity OGS × OGS → BOOLEAN
pré
post +Sn(o, o') ⇔ +Sn(Représentation(o), Représentation(o'))
hints
o, o' : OGS
OPERATION +St(o, o') *superposition totale entre deux objets géo-graphiques simples*
arity OGS × OGS → BOOLEAN
pré ¬(((Is-of-OGS_L(o) ∨ Is-of-OGS_POL(o)) ∧ Is-of-OGS_P(o')) ∨
(Is-of-OGS_POL(o) ∧ Is-of-OGS_L(o')))
post +St(o, o') ⇔ +St(Représentation(o), Représentation(o'))
hints
o, o' : OGS
OPERATION +Sp(o, o') *superposition partielle entre deux objets géo-graphiques simples*
arity OGS × OGS → BOOLEAN
pré ¬(Is-of-OGS_P(o))
post +Sp(o, o') ⇔ +Sp(Représentation(o), Représentation(o'))
hints
o, o' : OGS
OPERATION +V(o, o') *voisinage entre deux objets géo-graphiques simples*
arity OGS × OGS → BOOLEAN
pré Is-of-OGS_POL(o) ∨ Is-of-OGS_POL(o')
post +V(o, o') ⇔ V(Représentation(o), Représentation(o'))
hints
o, o' : OGS
OPERATION +Vt(o, o')*voisinage total entre deux objets géo-graphiques simples*
arity OGS × OGS → BOOLEAN
pré (Is-of-OGS_POL(o) ∨ Is-of-OGS_POL(o')) ∧ ¬(Is-of-OGS_P(o'))
post +Vt(o, o') ⇔ +Vt(Représentation(o), Représentation(o'))
hints o, o' : OGS
OPERATION +Vp(o, o') *voisinage partiel entre deux objets géo-graphiques simples*
arity OGS × OGS → BOOLEAN
pré (Is-of-OGS_POL(o) ∨ Is-of-OGS_POL(o')) ∧ ¬(Is-of-OGS_P(o))
post +Vp(o, o') ⇔ +Vp(Représentation(o), Représentation(o'))
hints o, o' : OGS
OPERATION +Vn(o, o') *voisinage nul entre deux objets géo-graphiques simples*
arity OGS × OGS → BOOLEAN
pré (Is-of-OGS_POL(o) ∨ Is-of-OGS_POL(o'))
post +Vn(o, o')⇔ ¬(+V(o, o'))
hints o, o' : OGS
* + les opérations -Vn, -Vt, -Vp, -Sn, -Sp, -Sp qui sont les négations des opérations ci-dessus*
END OGS

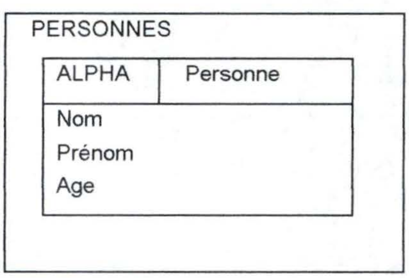
3.3.1.2. Les classes

Nous avons vu (chapitre 2) qu'une classe se composait d'objets, d'attributs logiques, et parfois de contraintes topologiques sur les instances des objets géo-graphiques contenus dans la classe. Les objets géo-graphiques et non géo-graphiques ne pouvant pas cohabiter dans une même classe et les classes composées d'objets non géo-graphiques n'ayant jamais de contraintes topologiques définies, nous avons choisi de séparer la définition d'une classe contenant des objets non géo-graphiques de celle d'une classe contenant des objets géo-graphiques.

Nous pouvons définir la classe contenant un objet non géo-graphique comme un cluster paramétré :

```
CLUSTER CLASSE[ONONG] *classe contenant un objet non géo-graphique*
 exports all
 is CP[Alogique : AT_LOG, Ens : SET[ONONG]]
 asserts
  ¬(Empty?(Ens(c)))
 hints
  c : CLASSE[ONONG]
-------------------------------------------------------------------
END CLASSE[ONONG]
```

Exemple de traduction d'une notation à une autre :




```
PERSONNES is CLASSE[PERSONNE]
asserts
  ∀ps : Is-of-(CP[])[Alogique(ps)]
hints ps : PERSONNES
PERSONNE is ONONG
asserts
  ∀ p : Is-of-(CP[Nom : STRING,
  Prénom : STRING, Age : INTEGER])[Alogique(p)]
hints p : PERSONNES
```

Nous avons vu que l'auteur individualisait une classe contenant un seul objet contenant une seule instance. Nous pouvons la définir par :

```
CLUSTER CLASSE1[ONONG] *classe contenant une seule instance d'objet*
 exports all
 is CLASSE[ONONG]
 asserts
Card(Ens(c)) = 1
 hints
  c : CLASSE1[ONONG]
-------------------------------------------------------------------
END CLASSE1[ONONG]
```

Nous venons de formaliser les classes contenant un objet non géo-graphique mais elles peuvent aussi en contenir plusieurs. Les objets appartenant à une même classe doivent avoir une sémantique homogène mais aucune restriction n'est faite sur leurs attributs.

Nous définissons une classe contenant deux objets non géo-graphiques par :

```
CLUSTER CLASSE[ONONG, ONONG'] *classe contenant deux objets*
 exports all
 is CP[Alogique : AT_LOG, Ens1 : SET[ONONG], Ens2 : SET[ONONG']]
 asserts
¬(Empty?(Ens1(c))) ∧ ¬(Empty?(Ens2(c)))
 hints
  c : CLASSE[ONONG, ONONG']
--------------------------------------------------------------
END CLASSE[ONONG, ONONG]
```

Formellement un cluster devrait être défini pour une classe possédant deux objets, puis trois objets, puis quatre objets,... Mais nous nous contentons de généraliser avec le cluster qui suit.

```
CLUSTER CLASSE[ONONG, ONONG,...,ONONG] *classe contenant plusieurs objets*
 exports all
 is CP[Alogique : AT_LOG, SET[ONONG], SET[ONONG],...,SET[ONONG]]
 asserts
  ...
 hints
--------------------------------------------------------------
END CLASSE[ONONG, ONONG,...,ONONG]
```

Comme pour la définition de la classe contenant un objet non géo-graphique nous allons définir la classe contenant un objet géo-graphique mais cette fois des contraintes topologiques peuvent être renseignées en plus dans la classe. Nous définissons aussi cette classe par un cluster paramétré.

CLASSE[OGS] is CP[AT_LOG, VOISPERMIS, SUPPERMIS, SUPOBLI, VOISOBLI, SET[OGS]]

ou

VOISOBLI is ENUM('+Vnobli', '+Vpobli', '+Vtobli', '-Vnobli', '-Vpobli', '-Vtobli','rien')

SUPOBLI is ENUM('+Snobli', '+Spobli', '+Stobli', '-Snobli', '-Spobli', '-Stobli','rien')

VOISPERMIS is ENUM('Vn,p', 'Vp,t', 'Vn,t', 'Vn,p,t', 'rien')

SUPPERMIS is ENUM('Sn,p', 'Sp,t', 'Sn,t', 'Sn,p,t', 'rien')

A propos des contraintes topologiques inscrites dans la classe il est important de faire quelques remarques avant de continuer.

Tout d'abord, il faut signaler que selon l'objet contenu dans la classe, seules certaines contraintes topologiques sont autorisées vu que toutes les contraintes topologiques ne sont pas définies pour tous les types d'objets. Par exemple si une classe contient un objet de type point, il ne peut y avoir des contraintes de voisinage puisqu'elles ne sont pas définies pour un objet de type point.

Ensuite, nous avons vu que toutes les contraintes topologiques ne sont pas symétriques (Vt, Vp, Sp, St). Dans les relations topologiques entre des objets de classes différentes (voir point quatre), une flèche indique le sens de la relation, par contre dans la classe aucune flèche n'est indiquée sur les contraintes topologiques. Nous pouvons donc en conclure que la relation doit être vérifiée dans les deux sens. Remarquons que cela restreint donc le nombre de relations qui peuvent être utilisées pour une classe. Par exemple si des objets de type point et de type polygone cohabitent dans la même classe les propriétés de Sp, St, Vp, Vt, ne peuvent pas être utilisées. Nous définirons formellement ce point dans les clusters.

De plus il faut remarquer que l'ensemble des contraintes topologiques pouvant être utilisé dans une classe ne dépend pas seulement du type de représentation géométrique des objets de la classe mais aussi des relations topologiques définies entre cette classe et une autre classe ou encore entre des objets de cette classe et des objets d'une autre classe. Il est très difficile de vérifier et de maintenir la cohérence d'un schéma CONGOO au niveau topologique car toutes les relations et contraintes topologiques définies dans un schéma sont interdépendantes. Nous ne tenons pas compte de cette interdépendance dans nos définitions formelles.

Enfin, signalons qu'il y a une redondance d'informations concernant les contraintes topologiques : en effet les contraintes permises peuvent directement être déduites des contraintes topologiques obligatoires.

CLUSTER CLASSE[OGS] *classe contenant un objet géo-graphique simple*
exports all
is CP[Alogique : AT_LOG, Vpermis : VOISPERMIS, Spermis : SUPPERMIS, Sobli : SUPOBLI, Vobli : VOISOBLI, Ens : SET[OGS]]

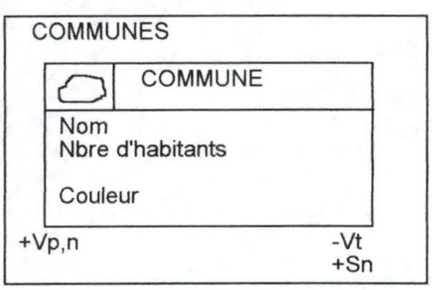
asserts
une classe contenant un objet géo-graphique de type point ne peut avoir de contraintes topologiques de voisinage et de superposition partielle
∀ o : o ∈ Ens(c) ∧ Is-of-OGS_P(o) ⇒ Vobli(c)= 'rien' ∧ Vpermis(c)= 'rien' ∧ Sobli(c) ≠ '+Spobli' ∧ Sobli(c) ≠ '-Spobli' ∧ (Spermis = 'Sn,t' ∨ Spermis = 'rien')
une classe contenant un objet géo-graphique de type point ne peut avoir de contraintes topologiques de voisinage
∀ o : o ∈ Ens(c) ∧ Is-of-OGS_L(o) ⇒ Vobli(c)= 'rien' ∧ Vpermis(c)= 'rien'
une contrainte topologique de voisinage nul dans une classe contenant un objet géo-graphique signifie que toutes les instances de cet objet ont un voisinage nul entre-elles
Vobli(c) = '+Vnobli' ⇒ (∀ o,o' : o ∈ Ens(c) ∧ o' ∈ Ens(c) ∧ o ≠ o' ⇒+Vn(o, o'))
*une contrainte topologique de voisinage total ....

Vobli(c) = '+Vtobli' $\Rightarrow$ ($\forall$ o,o' : o $\in$ Ens(c) $\wedge$ o' $\in$ Ens(c) $\wedge$ o $\neq$ o' $\Rightarrow$ +Vt(o, o'))
Vobli(c) = '+Vpobli' $\Rightarrow$ ($\forall$ o,o' : o $\in$ Ens(c) $\wedge$ o' $\in$ Ens(c) $\wedge$ o $\neq$ o' $\Rightarrow$ +Vp(o, o'))
Vobli(c) = '-Vnobli' $\Rightarrow$ ($\forall$ o,o' : o $\in$ Ens(c) $\wedge$ o' $\in$ Ens(c) $\wedge$ o $\neq$ o' $\Rightarrow$ -Vn(o, o'))
Vobli(c) = '-Vtobli' $\Rightarrow$ ($\forall$ o,o' : o $\in$ Ens(c) $\wedge$ o' $\in$ Ens(c) $\wedge$ o $\neq$ o' $\Rightarrow$ -Vt(o, o'))
Vobli(c) = '-Vpobli' $\Rightarrow$ ($\forall$ o,o' : o $\in$ Ens(c) $\wedge$ o' $\in$ Ens(c) $\wedge$ o $\neq$ o' $\Rightarrow$ -Vp(o, o'))
Sobli(c) = '+Snobli' $\Rightarrow$ ($\forall$ o,o' : o $\in$ Ens(c) $\wedge$ o' $\in$ Ens(c) $\wedge$ o $\neq$ o' $\Rightarrow$ +Sn(o, o'))
Sobli(c) = '+Stobli' $\Rightarrow$ ($\forall$ o,o' : o $\in$ Ens(c) $\wedge$ o' $\in$ Ens(c) $\wedge$ o $\neq$ o' $\Rightarrow$ +St(o, o'))
Sobli(c) = '+Spobli' $\Rightarrow$ ($\forall$ o,o' : o $\in$ Ens(c) $\wedge$ o' $\in$ Ens(c) $\wedge$ o $\neq$ o' $\Rightarrow$ +Sp(o, o'))
Sobli(c) = '-Snobli' $\Rightarrow$ ($\forall$ o,o' : o $\in$ Ens(c) $\wedge$ o' $\in$ Ens(c) $\wedge$ o $\neq$ o' $\Rightarrow$ -Sn(o, o'))
Sobli(c) = '-Stobli' $\Rightarrow$ ($\forall$ o,o' : o $\in$ Ens(c) $\wedge$ o' $\in$ Ens(c) $\wedge$ o $\neq$ o' $\Rightarrow$ -St(o, o'))
Sobli(c) = '-Spobli' $\Rightarrow$ ($\forall$ o,o' : o $\in$ Ens(c) $\wedge$ o' $\in$ Ens(c) $\wedge$ o $\neq$ o' $\Rightarrow$ -Sp(o, o'))

une contrainte topologique permise de voisinage nul et partiel dans une classe contenant un objet géo-graphique signifie que toutes les instances de cet objet ont un voisinage nul ou partiel entre-elles
Vpermis(c) = 'Vn,p' $\Rightarrow$ ($\forall$ o,o' : o $\in$ Ens(c) $\wedge$ o' $\in$ Ens(c) $\wedge$ o $\neq$ o' $\Rightarrow$ +Vn(o, o') $\vee$ +Vp(o, o'))
Vpermis(c) = 'Vp,t' $\Rightarrow$ ($\forall$ o,o' : o $\in$ Ens(c) $\wedge$ o' $\in$ Ens(c) $\wedge$ o $\neq$ o' $\Rightarrow$ +Vt(o, o') $\vee$ +Vp(o, o'))
Vpermis(c) = 'Vn,t' $\Rightarrow$ ($\forall$ o,o' : o $\in$ Ens(c) $\wedge$ o' $\in$ Ens(c) $\wedge$ o $\neq$ o' $\Rightarrow$ +Vn(o, o') $\vee$ +Vt(o, o'))
Vpermis(c) = 'Vn,p,t'
$\Rightarrow$ ($\forall$ o,o' : o $\in$ Ens(c) $\wedge$ o' $\in$ Ens(c) $\wedge$ o $\neq$ o' $\Rightarrow$ +Vn(o, o') $\vee$ +Vt(o, o') $\vee$ +Vp(o,o'))
Spermis(c) = 'Sn,p' $\Rightarrow$ ($\forall$ o,o' : o $\in$ Ens(c) $\wedge$ o' $\in$ Ens(c) $\wedge$ o $\neq$ o' $\Rightarrow$ +Sn(o, o') $\vee$ +Sp(o, o'))
Spermis(c) = 'Sp,t' $\Rightarrow$ ($\forall$ o,o' : o $\in$ Ens(c) $\wedge$ o' $\in$ Ens(c) $\wedge$ o $\neq$ o' $\Rightarrow$ +St(o, o') $\vee$ +Sp(o, o'))
Spermis(c) = 'Sn,t' $\Rightarrow$ ($\forall$ o,o' : o $\in$ Ens(c) $\wedge$ o' $\in$ Ens(c) $\wedge$ o $\neq$ o' $\Rightarrow$ +Sn(o, o') $\vee$ +St(o, o'))
Spermis(c) = 'Sn,p,t'
$\Rightarrow$ ($\forall$ o,o' : o $\in$ Ens(c) $\wedge$ o' $\in$ Ens(c) $\wedge$ o $\neq$ o' $\Rightarrow$ +Sn(o, o') $\vee$ +St(o, o') $\vee$ +Sp(o,o'))
une classe contient toujours une instance d'objet
$\neg$(Empty?(Ens(c)))
hints
o, o' : OGS
c : CLASSE[OGS]

END CLASSE[OGS]

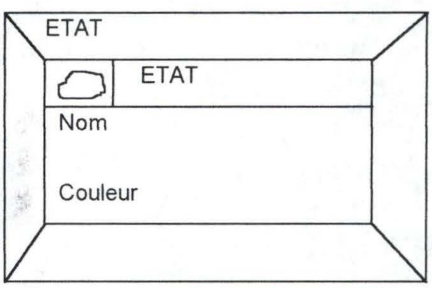
Exemple de traduction d'une notation à une autre :



COMMUNES is CLASSE[COMMUNE]
asserts
$\forall$cs : Is-of-(CP[])[Alogique(cs)] $\wedge$ Vpermis(cs) = '+Vp,n' $\wedge$ Spermis(cs) = 'rien' $\wedge$ Vobli(cs) = '-Vtobli' $\wedge$ Sobli(cs) = '+Sn'
hints cs : COMMUNES
COMMUNE is OGS_POL
asserts
$\forall$c : Is-of-(CP[Non : STRING, Nbre d'habitants : INTEGER])[Alogique(c)] $\wedge$
Is-of-(CP[Couleur :COULEUR])[Agraphique(c)]
hints c : COMMUNE

CLUSTER CLASSE1[OGS] *classe contenant une instance d'objet géo-graphique simple*
exports all
is CLASSE[OGS]
asserts
Card(Ens(c)) = 1
Vpermis(c) = 'rien' ∧ Spermis(c) = 'rien' ∧ Sobli(c) = 'rien' ∧ Vobli(c) = 'rien'
hints
c : CLASSE1[OGS]

END CLASSE1[OGS]



ETAT1 is CLASSE1[ETAT]
asserts
∀cs : Is-of-(CP[])[Alogique(cs)]
hints cs : ETAT1
ETAT is OGS_POL
asserts
∀e : Is-of-(CP[Nom : STRING][Alogique(e)] ∧ Is-of-(CP[Couleur : COULEUR])[Agraphique(e)]
hints e : ETAT

Nous venons de formaliser les classes contenant un objet géo-graphique mais elles peuvent aussi en contenir plusieurs comme pour les classes d'objets non géo-graphiques.

CLUSTER CLASSE[OGS, OGS'] *classe contenant deux objets géo-graphiques simples*
exports all
is CP[Alogique : AT_LOG, Vpermis : VOISPERMIS, Spermis : SUPPERMIS, Sobli : SUPOBLI, Vobli : VOISOBLI, Ens1 : SET[OGS], Ens2 : SET[OGS']]
asserts
*si les deux objets contenus dans la classe sont de type point alors il ne peut avoir de contrainte topologique de voisinage, de superposition partielle *
∀ o, o' : o ∈ Ens1(c) ∧ o' ∈ Ens2(c) ∧ Is-of-OGS_P(o) ∧ Is-of-OGS_P(o')⇒ Vobli(c)= 'rien' ∧ Vpermis(c)= 'rien' ∧ Sobli(c) ≠ '+Spobli' ∧ Sobli(c) ≠ '-Spobli' ∧ (Spermis = 'Sn,t' ∨ Spermis = 'rien')
si les deux objets ....
∀ o, o' : o ∈ Ens1(c) ∧ o' ∈ Ens2(c) ∧ Is-of-OGS_L(o) ∧ Is-of-OGS_L(o') ⇒ Vobli(c)= 'rien' ∧ Vpermis(c)= 'rien'
∀ o, o' : o ∈ Ens1(c) ∧ o' ∈ Ens2(c) ∧ Is-of-OGS_P(o) ∧ Is-of-OGS_L(o') ⇒ Vobli(c)= 'rien' ∧ Vpermis(c)= 'rien' ∧ (Sobli(c) = '+Snobli' ∨ Sobli(c) = '-Snobli')
∀ o, o' : o ∈ Ens1(c) ∧ o' ∈ Ens2(c) ∧ Is-of-OGS_P(o) ∧ Is-of-OGS_POL(o') ⇒ (Vobli(c)= '+Vnobli'∨Vobli(c)= '-Vnobli') ∧ Vpermis(c)= 'rien' ∧ (Sobli(c) = '+Snobli' ∨ Sobli(c) = '-Snobli')
∀ o, o' : o ∈ Ens1(c) ∧ o' ∈ Ens2(c) ∧ Is-of-OGS_POL(o) ∧ Is-of-OGS_L(o') ⇒ (Vobli(c)= '+Vnobli'∨Vobli(c)= '-Vnobli') ∧ Vpermis(c)= 'rien' ∧ (Sobli(c) = '+Snobli' ∨ Sobli(c) = '-Snobli')
la contrainte de voisinage nul d'une classe contenant deux objets signifie que toutes les instances des deux objets ont un voisinage nul entre-elles

$Vobli(c) = \text{'+Vnobli'} \Rightarrow (\forall o1,o2,o'1,o'2 : o1 \in Ens1(c) \wedge o'1 \in Ens1(c) \wedge o2 \in Ens2(c) \wedge o'2 \in Ens2(c) \wedge o1 \neq o2 \wedge o'1 \neq o'2 \Rightarrow +Vn(o1, o'1) \wedge +Vn(o1, o2) \wedge +Vn(o'1, o'2))$

la contrainte de voisinage total ...

$Vobli(c) = \text{'+Vtobli'} \Rightarrow (\forall o1,o2,o'1,o'2 : o1 \in Ens1(c) \wedge o'1 \in Ens1(c) \wedge o2 \in Ens2(c) \wedge o'2 \in Ens2(c) \wedge o1 \neq o2 \wedge o'1 \neq o'2 \Rightarrow +Vt(o1, o'1) \wedge +Vt(o1, o2) \wedge +Vt(o'1, o'2) \wedge +Vt(o'1, o1))$

$Vobli(c) = \text{'+Vpobli'} \Rightarrow (\forall o1,o2,o'1,o'2 : o1 \in Ens1(c) \wedge o'1 \in Ens1(c) \wedge o2 \in Ens2(c) \wedge o'2 \in Ens2(c) \wedge o1 \neq o2 \wedge o'1 \neq o'2 \Rightarrow +Vp(o1, o'1) \wedge +Vp(o1, o2) \wedge +Vp(o'1, o'2) \wedge +Vp(o'1, o1))$

$Vobli(c) = \text{'-Vnobli'} \Rightarrow (\forall o1,o2,o'1,o'2 : o1 \in Ens1(c) \wedge o'1 \in Ens1(c) \wedge o2 \in Ens2(c) \wedge o'2 \in Ens2(c) \wedge o1 \neq o2 \wedge o'1 \neq o'2 \Rightarrow -Vn(o1, o'1) \wedge -Vn(o1, o2) \wedge -Vn(o'1, o'2))$

$Vobli(c) = \text{'-Vtobli'} \Rightarrow (\forall o1,o2,o'1,o'2 : o1 \in Ens1(c) \wedge o'1 \in Ens1(c) \wedge o2 \in Ens2(c) \wedge o'2 \in Ens2(c) \wedge o1 \neq o2 \wedge o'1 \neq o'2 \Rightarrow -Vt(o1, o'1) \wedge -Vt(o1, o2) \wedge -Vt(o'1, o'2) \wedge -Vt(o'1, o1))$

$Vobli(c) = \text{'-Vpobli'} \Rightarrow (\forall o1,o2,o'1,o'2 : o1 \in Ens1(c) \wedge o'1 \in Ens1(c) \wedge o2 \in Ens2(c) \wedge o'2 \in Ens2(c) \wedge o1 \neq o2 \wedge o'1 \neq o'2 \Rightarrow -Vp(o1, o'1) \wedge -Vp(o1, o2) \wedge -Vp(o'1, o'2) \wedge -Vp(o'1, o1))$

$Sobli(c) = \text{'+Snobli'} \Rightarrow (\forall o1,o2,o'1,o'2 : o1 \in Ens1(c) \wedge o'1 \in Ens1(c) \wedge o2 \in Ens2(c) \wedge o'2 \in Ens2(c) \wedge o1 \neq o2 \wedge o'1 \neq o'2 \Rightarrow +Sn(o1, o'1) \wedge +Sn(o1, o2) \wedge +Sn(o'1, o'2))$

$Sobli(c) = \text{'+Stobli'} \Rightarrow (\forall o1,o2,o'1,o'2 : o1 \in Ens1(c) \wedge o'1 \in Ens1(c) \wedge o2 \in Ens2(c) \wedge o'2 \in Ens2(c) \wedge o1 \neq o2 \wedge o'1 \neq o'2 \Rightarrow +St(o1, o'1) \wedge +St(o1, o2) \wedge +St(o'1, o'2) \wedge +St(o'1, o1))$

$Sobli(c) = \text{'+Spobli'} \Rightarrow (\forall o1,o2,o'1,o'2 : o1 \in Ens1(c) \wedge o'1 \in Ens1(c) \wedge o2 \in Ens2(c) \wedge o'2 \in Ens2(c) \wedge o1 \neq o2 \wedge o'1 \neq o'2 \Rightarrow +Sp(o1, o'1) \wedge +Sp(o1, o2) \wedge +Sp(o'1, o'2) \wedge +Sp(o'1, o1))$

$Sobli(c) = \text{'-Snobli'} \Rightarrow (\forall o1,o2,o'1,o'2 : o1 \in Ens1(c) \wedge o'1 \in Ens1(c) \wedge o2 \in Ens2(c) \wedge o'2 \in Ens2(c) \wedge o1 \neq o2 \wedge o'1 \neq o'2 \Rightarrow -Sn(o1, o'1) \wedge -Sn(o1, o2) \wedge -Sn(o'1, o'2))$

$Sobli(c) = \text{'-Stobli'} \Rightarrow (\forall o1,o2,o'1,o'2 : o1 \in Ens1(c) \wedge o'1 \in Ens1(c) \wedge o2 \in Ens2(c) \wedge o'2 \in Ens2(c) \wedge o1 \neq o2 \wedge o'1 \neq o'2 \Rightarrow -St(o1, o'1) \wedge -St(o1, o2) \wedge -St(o'1, o'2) \wedge -St(o'1, o1))$

$Sobli(c) = \text{'-Spobli'} \Rightarrow (\forall o1,o2,o'1,o'2 : o1 \in Ens1(c) \wedge o'1 \in Ens1(c) \wedge o2 \in Ens2(c) \wedge o'2 \in Ens2(c) \wedge o1 \neq o2 \wedge o'1 \neq o'2 \Rightarrow -Sp(o1, o'1) \wedge -Sp(o1, o2) \wedge -Sp(o'1, o'2) \wedge -St(o'1, o1))$

une contrainte topologique permise de voisinage nul et partiel dans une classe contenant deux objets géo-graphiques signifie que toutes les instances de ces objets ont un voisinage nul ou partiel entre-elles

$Vpermis(c) = \text{'Vn,p'} \Rightarrow (\forall o1,o2,o'1,o'2 : o1 \in Ens1(c) \wedge o'1 \in Ens1(c) \wedge o2 \in Ens2(c) \wedge o'2 \in Ens2(c) \wedge o1 \neq o2 \wedge o'1 \neq o'2 \Rightarrow -Vt(o1, o'1) \wedge -Vt(o1, o2) \wedge -Vt(o'1, o'2) \wedge -Vt(o'1, o1))$

$Vpermis(c) = \text{'Vp,t'} \Rightarrow (\forall o1,o2,o'1,o'2 : o1 \in Ens1(c) \wedge o'1 \in Ens1(c) \wedge o2 \in Ens2(c) \wedge o'2 \in Ens2(c) \wedge o1 \neq o2 \wedge o'1 \neq o'2 \Rightarrow -Vn(o1, o'1) \wedge -Vn(o1, o2) \wedge -Vn(o'1, o'2))$

$Vpermis(c) = \text{'Vn,t'} \Rightarrow (\forall o1,o2,o'1,o'2 : o1 \in Ens1(c) \wedge o'1 \in Ens1(c) \wedge o2 \in Ens2(c) \wedge o'2 \in Ens2(c) \wedge o1 \neq o2 \wedge o'1 \neq o'2 \Rightarrow -Vp(o1, o'1) \wedge -Vp(o1, o2) \wedge -Vp(o'1, o'2) \wedge -Vp(o'1, o1))$

$Vpermis(c) = \text{'Vn,p,t'} \Rightarrow (\forall o1,o2,o'1,o'2 : o1 \in Ens1(c) \wedge o'1 \in Ens1(c) \wedge o2 \in Ens2(c) \wedge o'2 \in Ens2(c) \wedge o1 \neq o2 \wedge o'1 \neq o'2 \Rightarrow (+Vn(o1 , o'1) \vee +Vp(o1, o'1) \vee +Vt(o1\ o'1) \wedge (+Vn(o1 , o2) \vee +Vp(o1, o2) \vee +Vt(o1\ o2) \wedge (+Vn(o'1 , o'2) \vee +Vp(o'1, o'2) \vee +Vt(o'1, o'2)))$

$Spermis(c) = \text{'Sn,p'} \Rightarrow (\forall o1,o2,o'1,o'2 : o1 \in Ens1(c) \wedge o'1 \in Ens1(c) \wedge o2 \in Ens2(c) \wedge o'2 \in Ens2(c) \wedge o1 \neq o2 \wedge o'1 \neq o'2 \Rightarrow -St(o1, o'1) \wedge -St(o1, o2) \wedge -St(o'1, o'2) \wedge -St(o'1, o1))$

$Spermis(c) = \text{'Sp,t'} \Rightarrow (\forall o1,o2,o'1,o'2 : o1 \in Ens1(c) \wedge o'1 \in Ens1(c) \wedge o2 \in Ens2(c) \wedge o'2 \in Ens2(c) \wedge o1 \neq o2 \wedge o'1 \neq o'2 \Rightarrow -Sn(o1, o'1) \wedge -Sn(o1, o2) \wedge -Sn(o'1, o'2))$

$Spermis(c) = \text{'Sn,t'} \Rightarrow (\forall o1,o2,o'1,o'2 : o1 \in Ens1(c) \wedge o'1 \in Ens1(c) \wedge o2 \in Ens2(c) \wedge o'2 \in Ens2(c) \wedge o1 \neq o2 \wedge o'1 \neq o'2 \Rightarrow -Sp(o1, o'1) \wedge -Sp(o1, o2) \wedge -Sp(o'1, o'2) \wedge -St(o'1, o1))$

$Spermis(c) = \text{'Sn,p,t'} \Rightarrow (\forall o1,o2,o'1,o'2 : o1 \in Ens1(c) \wedge o'1 \in Ens1(c) \wedge o2 \in Ens2(c) \wedge o'2 \in Ens2(c) \wedge o1 \neq o2 \wedge o'1 \neq o'2 \Rightarrow (+Sn(o1 , o'1) \vee +Sp(o1, o'1) \vee +St(o1\ o'1) \wedge (+Sn(o1 , o2) \vee +Sp(o1, o2) \vee +St(o1\ o2) \wedge (+Sn(o'1 , o'2) \vee +Sp(o'1, o'2) \vee +St(o'1, o'2)))$

une classe de deux objets contient toujours une instance de chaque objet
¬(Empty?(Ens1(c))) ∧ ¬(Empty?(Ens2(c)))
hints
o1, o2, o'1, o'2 : OGS
c : CLASSE[OGS, OGS]
- -
END CLASSE[OGS, OGS]

CLUSTER CLASSE[OGS, OGS,...,OGS] *classe contenant plusieurs objets géo-graphiques *
exports all
is CP[Alogique : AT_LOG, Vpermis : VOISPERMIS, Spermis : SUPPERMIS, Sobli : SUPOBLI, Vobli : VOISOBLI, Ens1 : SET[OGS], Ens2 : SET[OGS],..., Ensn : SET[OGS]]
asserts
...
hints
...
- -
END CLASSE[OGS, OGS,...,OGS]

3.3.1.3. Les relations logiques

Nous avons vu dans le chapitre deux que le modèle CONGOO supporte différents types de relations. Nous allons étudier tout d'abord les relations logiques.

Les relations logiques entre objets peuvent être définies comme un cluster paramétré ou les deux paramètres formels sont des objets.

REL_LOG[OBJET, OBJET']
is CP[card1 : CARD, card2 : CARD, SET[CP[OBJET, OBJET']], SET[OBJET], SET[OBJET']]

Dans la présentation du modèle, il est indiqué que des contraintes de cardinalité doivent accompagner les relations logiques entre objets. Toutefois dans certains schémas présentés (par exemple le schéma de la figure 3.1), les relations logiques entre deux objets ne possèdent pas nécessairement de contraintes de cardinalité. Nous avons pu constater après une brève analyse que cela signifie que toutes les instances d'un objet participent à cette relation avec toutes les instances de l'autre objet et, réciproquement. Pour tenir compte de ce cas, nous avons décidé d'ajouter la notation 'rien' à l'ensemble des notations des contraintes de cardinalités définies.

CARD is ENUM('0-n', '0-1', '1-1', '1-n', 'rien')

CLUSTER REL_LOG[OBJET, OBJET'] *relation logique entre deux objets*
exports all
is CP[Card1 : CARD, Card2 : CARD, Rel : SET[CP[OBJET, OBJET']], Cens1 : SET[OBJET],

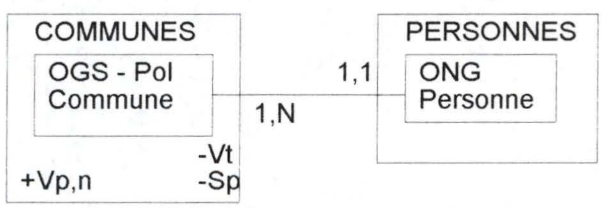
Cens2 : SET[OBJET]]
asserts
∀ o, o' : In(<o, o'>, Rel(r)) ⇒ o ∈ Cens1(r) ∧ o' ∈ Cens2(r)
Card1(r) = '1-n' ⇒ ∀ o ∃ o' : In(<o, o'>, Rel(r)) ∧ o ∈ Cens1(r) ∧ o' ∈ Cens2(r)
Card1(r) = '0-1' ⇒ (∀ o, o'1, o'2 : o ∈ Cens1(r) ∧ o'1 ∈ Cens2(r)∧ o'2 ∈ Cens2(r) ∧ In(<o, o'1>, Rel(r)) ∧ In(<o, o'2>, Rel(r))⇒ o'1 = o'2)
Card1(r) = '1-1' ⇒ (∀ o ∃ o'1 : o ∈ Cens1(r) ∧ o'1 ∈ Cens2(r)∧ o'2 ∈ Cens2(r) ∧ In(<o, o'1>, Rel(r)) ∧ In(<o, o'2>, Rel(r)) ⇒ o'1 = o'2))
Card2(r) = '1-n' ⇒ ∀ o' ∃ o : o ∈ Cens1(r) ∧ o' ∈ Cens2(r) ∧ In(<o, o'>, Rel(r))
Card2(r) = '0-1' ⇒ (∀ o1, o', o2 : o1 ∈ Cens1(r) ∧ o2 ∈ Cens1(r)∧ o' ∈ Cens2(r) ∧ In(<o1, o'>, Rel(r)) ∧ In(<o2, o'>, Rel(r))⇒ o1 = o2)
Card2(r) = '1-1' ⇒ (∀ o' ∃ o1 : o ∈ Cens1(r) ∧ o'1 ∈ Cens2(r)∧ In(<o1, o'>, Rel(r)) ∧ In(<o2, o'>, Rel(r)) ⇒ o1 = o2))
Card1(r) = 'rien' ⇔ Card2(r) = 'rien'
Card1(r) = 'rien' ∨ Card2(r) = 'rien' ⇒ ∀o, o' : o ∈ Cens1(r) ∧ Card(Filter(Rel(r), Sel1(Rel(r)i) = o))) = Card (Cens2(r)) ∧ Card(Filter(Rel(r), Sel2(Rel(r)i) = o'))) = Card (Cens1(r)) ∧ o' ∈ Cens2(r)

hints
r : REL_LOG[OBJET,OBJET']
o, o1, o2 : OBJET
o', o'1, o'2 : OBJET'

- -

END REL_LOG[OBJET, OBJET']

Exemple de traduction :



SCHEMA = CP[C : COMMUNES, P : PERSONNES, R : REL_LOG_1-n_1-1[COMMUNE, PERSONNE]]
asserts : ∀s : Ens(C(s)) = Cens1(R(s)) ∧ Ens(P(s)) = Cens2(R(s))
hints : s : SCHEMA
REL_LOG_1-n_1-1[COMMUNE, PERSONNE] is REL_LOG[COMMUNE, PERSONNE]
asserts : ∀r : Card1(r) = '1-n' ∧ Card2(r) = '1-1'
hints r: REL_LOG_1-n_1-1[COMMUNE, PERSONNE]
COMMUNES is CLASSE[COMMUNE]
asserts : ∀cs : Vobli(cs) = '-Vt' ∧ Vpermis(cs) = '+Vp,n' ∧ Sobli(cs) = '-Sp' ∧ Spermis(cs) = 'rien'
hints : cs : COMMUNES
PERSONNES is CLASSE[PERSONNE]
COMMUNE is OGS-Pol
PERSONNE is ONG

D'après l'auteur les relations logiques peuvent exister avec des contraintes de cardinalités entre objets, classes, couches, sous-couches et leurs combinaisons. Nous pouvons donc penser qu'il faut définir un paramètre formel plus général mais vu qu'un schéma CONGOO est un produit cartésien de plusieurs classes, couches, sous-couches et de relations, il n'est pas possible d'avoir une relation entre classes, couches et sous-couches où il y a plus d'une instance de ces éléments qui participe à la relation. Nous pensons donc qu'il y a une erreur à ce niveau. Remarquons que pour les autres catégories de relations (relations de structure, relations topologiques) il est indiqué que seules les relations entre objets peuvent avoir des cardinalités et pas entre les classes, les couches ou les sous-couches. Nous allons appliquer la solution de ne pas définir de cardinalité entre les classes, les couches, les sous-couches et leurs combinaisons comme pour les autres catégories de relations. Auparavant vu qu'il existe différents clusters de classes suivant le nombre d'objets qu'ils contiennent nous allons définir un nouveau cluster classe plus général qui sera utilisé comme paramètre formel.

```
CLUSTER CLASSE
 exports all
 is UNION[CLASSE1[OGS], CLASSE1[ONONG], CLASSE[OGS], CLASSE[ONONG],
CLASSE[OGS, OGS],..., CLASSE[ONONG, ONONG,...,ONONG]]
 asserts
 hints
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
END CLASSE
```

Nous pouvons définir une relation logique entre deux classes par :

```
CLUSTER REL_CL[CLASSE, CLASSE] *relation logique entre deux classes*
 exports all
 is CP[CLASSE, CLASSE]
 asserts
 hints
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
END REL_CL[CLASSE, CLASSE]
```

```
CLUSTER REL_LOG[CLASSE, OBJET] *relation logique entre un objet et une classe*
 exports all
 is CP[Card1 : CARD, Card2 : CARD, Rel : SET[CP[CLASSE, OBJET]], Cens :
SET[OBJET]]
 asserts
Card1(r) ≠ '0-1' ∧ Card1(r) ≠ '0-N' ∧ Card2(r) ≠ '0-N' ∧ Card2(r) ≠ '1-N'
∀ o : In(<c, o>, Rel(r)) ⇒ o ∈ Cens(r)
Card1(r) = '1-n' ⇒ ∃ o : In(<c, o>, Rel(r)) ∧ o ∈ Cens(r)
Card1(r) = '1-1' ⇒ (∃ o : o ∈ Cens(r) ∧ o' ∈ Cens(r) ∧ In(<c, o>, Rel(r)) ∧ In(<c, o'>, Rel(r))
⇒ o = o'))
Card2(r) = '1-1' ⇒ (∀ o : o ∈ Cens(r) ⇒ In(<c, o>, Rel(r)))
```

Card1(r) = 'rien' ⇔ Card2(r) = 'rien'
Card1(r) = 'rien' ∨ Card2(r) = 'rien' ⇒ (∀o : o ∈ Cens(r) ⇒ In(<c, o>, Rel(r)))
hints
r : REL_LOG[CLASSE,OBJET']
o, o': OBJET

END REL_LOG[CLASSE, OBJET]

Nous discuterons au point 5 les relations logiques concernant les couches et sous-couches.

3.3.1.4. Les relations topologiques

D'après l'auteur les relations topologiques peuvent être définies entre des objets géographiques, des classes, des sous-couches, des couches et leurs combinaisons. Remarquons que les classes, les sous-couches et couches doivent nécessairement contenir des objets géographiques pour que ces relations aient un sens.

Deux nouveaux types sont définis pour reprendre les différentes notations que l'on peut trouver dans la relation. Remarquons que le sens de la réalisation de la propriété topologique est indiqué par une flèche quand cette propriété n'est pas symétrique.

VOISREL1 is ENUM('+Vn', '-Vn', '←+Vt', '←-Vt', '←+Vp', '←-Vp', 'rien')
VOISREL2 is ENUM('+Vt→', '-Vt→', '+Vp→', '-Vp→', 'rien')
SUPREL1 is ENUM('+Sn', '-Sn', '←+St', '←-St', '←+Sp', '←-Sp', 'rien')
SUPREL2 is ENUM('-St→', '+St→', '+Sp→', '-Sp→', 'rien')

Nous pouvons définir les relations topologiques entre des objets géo-graphiques simples comme ceci :

CLUSTER REL_TOPO[OGS, OGS'] *relations topologiques entre deux objets géographiques simples*
exports all
is CP[Card1 : CARD, Card2 : CARD, Srel1 : SUPREL1, Vrel1 : VOISREL1, Srel2 : SUPREL2, Vrel2 : VOISREL2, Rel : SET[CP[OGS, OGS']], Cens1 : SET[OGS], Cens2 : SET[OGS]]
asserts
∀ o, o' : o ∈ Cens1(r) ∧ o' ∈ Cens2(r) ∧ Is-of-OGS_P(o) ∧ Is-of-OGS_P(o')⇒ Vrel1(r)= 'rien' ∧Vrel2(r)= 'rien' ∧ Srel1(r) ≠ '←+Sp' ∧ Srel1(r) ≠ '←-Sp' ∧ Srel2(r) ≠ '+Sp→' ∧ Srel2(r) ≠ '-Sp→'
∀ o, o' : o ∈ Cens1(r) ∧ o' ∈ Cens2(r) ∧ Is-of-OGS_L(o) ∧ Is-of-OGS_L(o') ⇒ Vrel1(r) = 'rien' ∧ Vrel2(r) = 'rien'
∀ o, o' : o ∈ Cens1(r) ∧ o' ∈ Cens2(r) ∧ Is-of-OGS_P(o) ∧ Is-of-OGS_L(o') ⇒ Vrel1(r)= 'rien' ∧ Vrel2(r) = 'rien' ∧ Srel1(r) ≠ '←+St' ∧ Srel1(r) ≠ '←-St' ∧ Srel2(r) ≠ '+Sp→' ∧ Srel2(r) ≠ '-Sp→'
∀ o, o' : o ∈ Cens1(r) ∧ o' ∈ Cens2(r) ∧ Is-of-OGS_L(o) ∧ Is-of-OGS_P(o') ⇒ Vrel1(r)= 'rien' ∧ Vrel2(r) = 'rien' ∧ Srel1(r) ≠ '←+Sp' ∧ Srel1(r) ≠ '←-Sp' ∧ Srel2(r) ≠ '+St→' ∧ Srel2(r) ≠ '-St→'

$\forall o, o' : o \in \mathrm{Cens1}(r) \wedge o' \in \mathrm{Cens2}(r) \wedge \text{Is-of-OGS\_P}(o) \wedge \text{Is-of-OGS\_POL}(o') \Rightarrow \mathrm{Vrel1}(r) \neq \text{'}\leftarrow\text{+Vt'} \wedge \mathrm{Vrel1}(r) \neq \text{'}\leftarrow\text{-Vt'} \wedge \mathrm{Vrel2}(r) \neq \text{'+Vp}\rightarrow\text{'} \wedge \mathrm{Vrel2}(r) \neq \text{'-Vp}\rightarrow\text{'} \wedge \mathrm{Srel1}(r) \neq \text{'}\leftarrow\text{+St'} \wedge \mathrm{Srel1}(r) \neq \text{'}\leftarrow\text{-St'} \wedge \mathrm{Srel2}(r) \neq \text{'+Sp}\rightarrow\text{'} \wedge \mathrm{Srel2}(r) \neq \text{'-Sp}\rightarrow\text{'}$

$\forall o, o' : o \in \mathrm{Cens1}(r) \wedge o' \in \mathrm{Cens2}(r) \wedge \text{Is-of-OGS\_POL}(o) \wedge \text{Is-of-OGS\_P}(o') \Rightarrow \mathrm{Vrel1}(r) \neq \text{'}\leftarrow\text{+Vp'} \wedge \mathrm{Vrel1}(r) \neq \text{'}\leftarrow\text{-Vp'} \wedge \mathrm{Vrel2}(r) \neq \text{'+Vt}\rightarrow\text{'} \wedge \mathrm{Vrel2}(r) \neq \text{'-Vt}\rightarrow\text{'} \wedge \mathrm{Srel1}(r) \neq \text{'}\leftarrow\text{+Sp'} \wedge \mathrm{Srel1}(r) \neq \text{'}\leftarrow\text{-Sp'} \wedge \mathrm{Srel2}(r) \neq \text{'+St}\rightarrow\text{'} \wedge \mathrm{Srel2}(r) \neq \text{'-St}\rightarrow\text{'}$

$\forall o, o' : o \in \mathrm{Cens1}(r) \wedge o' \in \mathrm{Cens2}(r) \wedge \text{Is-of-OGS\_L}(o) \wedge \text{Is-of-OGS\_POL}(o') \Rightarrow \mathrm{Srel1}(r) \neq \text{'}\leftarrow\text{+St'} \wedge \mathrm{Srel1}(r) \neq \text{'}\leftarrow\text{-St'}$

$\forall o, o' : o \in \mathrm{Cens1}(r) \wedge o' \in \mathrm{Cens2}(r) \wedge \text{Is-of-OGS\_POL}(o) \wedge \text{Is-of-OGS\_L}(o') \Rightarrow \mathrm{Srel2}(r) \neq \text{'+St}\rightarrow\text{'} \wedge \mathrm{Srel2}(r) \neq \text{'-St}\rightarrow\text{'}$

$\forall o, o' : \mathrm{In}(\langle o, o'\rangle, \mathrm{Rel}(r)) \Rightarrow o \in \mathrm{Cens1}(r) \wedge o' \in \mathrm{Cens2}(r)$

$\mathrm{Card1}(r) = \text{'1-n'} \Rightarrow \forall o\ \exists o' : \mathrm{In}(\langle o, o'\rangle, \mathrm{Rel}(r)) \wedge o \in \mathrm{Cens1}(r) \wedge o' \in \mathrm{Cens2}(r)$

$\mathrm{Card1}(r) = \text{'0-1'} \Rightarrow (\forall o, o'1, o'2 : o \in \mathrm{Cens1}(r) \wedge o'1 \in \mathrm{Cens2}(r) \wedge o'2 \in \mathrm{Cens2}(r) \wedge \mathrm{In}(\langle o, o'1\rangle, \mathrm{Rel}(r)) \wedge \mathrm{In}(\langle o, o'2\rangle, \mathrm{Rel}(r)) \Rightarrow o'1 = o'2)$

$\mathrm{Card1}(r) = \text{'1-1'} \Rightarrow (\forall o\ \exists o'1 : o \in \mathrm{Cens1}(r) \wedge o'1 \in \mathrm{Cens2}(r) \wedge o'2 \in \mathrm{Cens2}(r) \wedge \mathrm{In}(\langle o, o'1\rangle, \mathrm{Rel}(r)) \wedge \mathrm{In}(\langle o, o'2\rangle, \mathrm{Rel}(r)) \Rightarrow o'1 = o'2))$

$\mathrm{Card2}(r) = \text{'1-n'} \Rightarrow \forall o'\ \exists o : o \in \mathrm{Cens1}(r) \wedge o' \in \mathrm{Cens2}(r) \wedge \mathrm{In}(\langle o, o'\rangle, \mathrm{Rel}(r))$

$\mathrm{Card2}(r) = \text{'0-1'} \Rightarrow (\forall o1, o', o2 : o1 \in \mathrm{Cens1}(r) \wedge o2 \in \mathrm{Cens1}(r) \wedge o' \in \mathrm{Cens2}(r) \wedge \mathrm{In}(\langle o1, o'\rangle, \mathrm{Rel}(r)) \wedge \mathrm{In}(\langle o2, o'\rangle, \mathrm{Rel}(r)) \Rightarrow o1 = o2)$

$\mathrm{Card2}(r) = \text{'1-1'} \Rightarrow (\forall o'\ \exists o1 : o \in \mathrm{Cens1}(r) \wedge o'1 \in \mathrm{Cens2}(r) \wedge \mathrm{In}(\langle o1, o'\rangle, \mathrm{Rel}(r)) \wedge \mathrm{In}(\langle o2, o'\rangle, \mathrm{Rel}(r)) \Rightarrow o1 = o2))$

$\mathrm{Card1}(r) = \text{'rien'} \Leftrightarrow \mathrm{Card2}(r) = \text{'rien'}$

$\mathrm{Card1}(r) = \text{'rien'} \vee \mathrm{Card2}(r) = \text{'rien'} \Rightarrow \forall o, o' : o \in \mathrm{Cens1}(r) \wedge \mathrm{Card}(\mathrm{Filter}(\mathrm{Rel}(r), \mathrm{Sel1}(\mathrm{Rel}(r)i) = o))) = \mathrm{Card}\ (\mathrm{Cens2}(r)) \wedge \mathrm{Card}(\mathrm{Filter}(\mathrm{Rel}(r), \mathrm{Sel2}(\mathrm{Rel}(r)i) = o'))) = \mathrm{Card}\ (\mathrm{Cens1}(r)) \wedge o' \in \mathrm{Cens2}(r)$

$\mathrm{Vrel1}(r) = \text{'+Vn'} \Rightarrow (\forall o,o' : \mathrm{In}(\langle o, o'\rangle, \mathrm{Rel}(r)) \Leftrightarrow +\mathrm{Vn}(o, o'))$

$\mathrm{Vrel1}(r) = \text{'-Vn'} \Rightarrow (\forall o,o' : \mathrm{In}(\langle o, o'\rangle, \mathrm{Rel}(r)) \Leftrightarrow -\mathrm{Vn}(o, o'))$

$\mathrm{Vrel2}(r) = \text{'+Vt}\rightarrow\text{'} \Rightarrow (\forall o,o' : \mathrm{In}(\langle o, o'\rangle, \mathrm{Rel}(r)) \Leftrightarrow +\mathrm{Vt}(o, o'))$

$\mathrm{Vrel1}(r) = \text{'}\leftarrow\text{+Vt'} \Rightarrow (\forall o,o' : \mathrm{In}(\langle o, o'\rangle, \mathrm{Rel}(r)) \Leftrightarrow +\mathrm{Vt}(o', o))$

$\mathrm{Vrel2}(r) = \text{'-Vt}\rightarrow\text{'} \Rightarrow (\forall o,o' : \mathrm{In}(\langle o, o'\rangle, \mathrm{Rel}(r)) \Leftrightarrow -\mathrm{Vt}(o, o'))$

$\mathrm{Vrel1}(r) = \text{'}\leftarrow\text{-Vt'} \Rightarrow (\forall o,o' : \mathrm{In}(\langle o, o'\rangle, \mathrm{Rel}(r)) \Leftrightarrow -\mathrm{Vt}(o', o))$

$\mathrm{Vrel2}(r) = \text{'+Vp}\rightarrow\text{'} \Rightarrow (\forall o,o' : \mathrm{In}(\langle o, o'\rangle, \mathrm{Rel}(r)) \Leftrightarrow +\mathrm{Vp}(o, o'))$

$\mathrm{Vrel1}(r) = \text{'}\leftarrow\text{+Vp'} \Rightarrow (\forall o,o' : \mathrm{In}(\langle o, o'\rangle, \mathrm{Rel}(r)) \Leftrightarrow +\mathrm{Vp}(o', o))$

$\mathrm{Vrel2}(r) = \text{'-Vp}\rightarrow\text{'} \Rightarrow (\forall o,o' : \mathrm{In}(\langle o, o'\rangle, \mathrm{Rel}(r)) \Leftrightarrow -\mathrm{Vp}(o, o'))$

$\mathrm{Vrel1}(r) = \text{'}\leftarrow\text{-Vp'} \Rightarrow (\forall o,o' : \mathrm{In}(\langle o, o'\rangle, \mathrm{Rel}(r)) \Leftrightarrow -\mathrm{Vp}(o', o))$

$\mathrm{Srel1}(r) = \text{'+Sn'} \Rightarrow (\forall o,o' : \mathrm{In}(\langle o, o'\rangle, \mathrm{Rel}(r)) \Leftrightarrow +\mathrm{Sn}(o, o'))$

$\mathrm{Srel1}(r) = \text{'-Sn'} \Rightarrow (\forall o,o' : \mathrm{In}(\langle o, o'\rangle, \mathrm{Rel}(r)) \Leftrightarrow -\mathrm{Sn}(o, o'))$

$\mathrm{Srel2}(r) = \text{'+St}\rightarrow\text{'} \Rightarrow (\forall o,o' : \mathrm{In}(\langle o, o'\rangle, \mathrm{Rel}(r)) \Leftrightarrow +\mathrm{St}(o, o'))$

$\mathrm{Srel1}(r) = \text{'}\leftarrow\text{+St'} \Rightarrow (\forall o,o' : \mathrm{In}(\langle o, o'\rangle, \mathrm{Rel}(r)) \Leftrightarrow +\mathrm{St}(o', o))$

$\mathrm{Srel2}(r) = \text{'-St}\rightarrow\text{'} \Rightarrow (\forall o,o' : \mathrm{In}(\langle o, o'\rangle, \mathrm{Rel}(r)) \Leftrightarrow -\mathrm{St}(o, o'))$

$\mathrm{Srel1}(r) = \text{'}\leftarrow\text{-St'} \Rightarrow (\forall o,o' : \mathrm{In}(\langle o, o'\rangle, \mathrm{Rel}(r)) \Leftrightarrow -\mathrm{St}(o', o))$

$\mathrm{Srel2}(r) = \text{'+Sp}\rightarrow\text{'} \Rightarrow (\forall o,o' : \mathrm{In}(\langle o, o'\rangle, \mathrm{Rel}(r)) \Leftrightarrow +\mathrm{Sp}(o, o'))$

$\mathrm{Srel1}(r) = \text{'}\leftarrow\text{+Sp'} \Rightarrow (\forall o,o' : \mathrm{In}(\langle o, o'\rangle, \mathrm{Rel}(r)) \Leftrightarrow +\mathrm{Sp}(o', o))$

$\mathrm{Srel2}(r) = \text{'-Sp}\rightarrow\text{'} \Rightarrow (\forall o,o' : \mathrm{In}(\langle o, o'\rangle, \mathrm{Rel}(r)) \Leftrightarrow -\mathrm{Sp}(o, o'))$

$\mathrm{Srel1}(r) = \text{'}\leftarrow\text{-Sp'} \Rightarrow (\forall o,o' : \mathrm{In}(\langle o, o'\rangle, \mathrm{Rel}(r)) \Leftrightarrow -\mathrm{Sp}(o', o))$

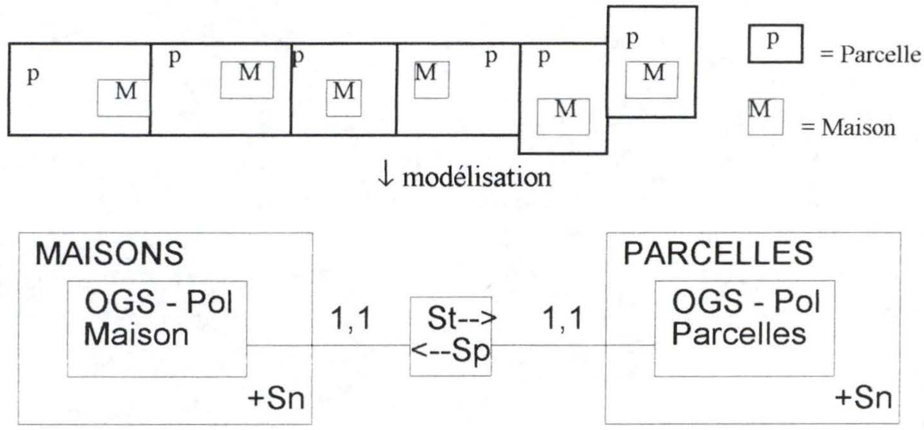
hints

r : REL_TOPO[OGS, OGS']
o, o1, o2 : OGS
o', o'1, o'2 : OGS'

- -

END REL_TOPO[OGS, OGS']

Exemple de traduction de notation :



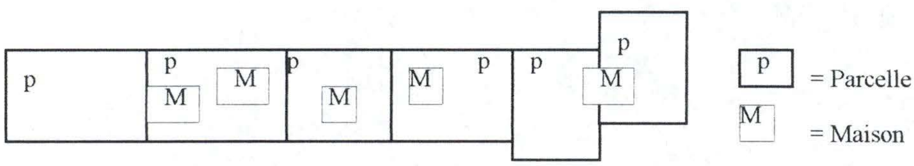
SCHEMA = CP[M : MAISONS, P : PARCELLES, R : REL_TOPO_1-1_1-1[MAISON, PARCELLE]]
asserts : ∀s : Ens(M(s)) = Cens1(R(s)) ∧ Ens(P(s)) = Cens2(R(s))
hints : s : SCHEMA
REL_TOPO_1-1_1-1[MAISON, PARCELLE] is REL_TOPO[MAISON, PARCELLE]
asserts : ∀r : Card1(r) = '1-1' ∧ Card2(r) = '1-1'∧ Vrel1(r) = 'rien' ∧ Vrel2(r) = 'rien' ∧ Srel1(r) = '←Sp' ∧ Srel2(r) = 'St→'
hints r: REL_TOPO_1-1_1-1[MAISON, PARCELLE]
MAISONS is CLASSE[MAISON]
asserts : ∀cs : Vobli(cs) = 'rien' ∧ Vpermis(cs) = 'rien' ∧ Sobli(cs) = '+Sn' ∧ Spermis(cs) = 'rien'
hints : cs : MAISONS
MAISON is OGS-POL
PARCELLES is CLASSE[PARCELLE]
asserts : ∀cs : Vobli(cs) = 'rien' ∧ Vpermis(cs) = 'rien' ∧ Sobli(cs) = '+Sn' ∧ Spermis(cs) = 'rien'
hints : cs : PARCELLES
PARCELLE is OGS-Pol

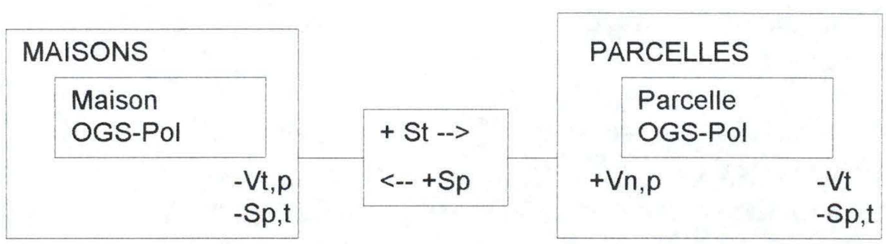
Rappelons qu'il y a une influence de la contrainte topologique sur les objets marqués dans les classes et la propriété topologique qui peut être définie par une relation entre ces deux objets. Par exemple si nous avons une classe d'objets de type point avec une contrainte de superposition nulle et une autre classe d'objets de type point avec une contrainte de superposition totale, il est impossible d'avoir une relation topologique de superposition totale avec une cardinalité 1-1 de chaque côté. Dans le cluster ci-dessus, pour définir les invariants nous n'avons pas tenu compte, des contraintes topologiques exprimées dans la classe. En fait il est très difficile de maintenir la cohérence entre toutes les propriétés topologiques (contrainte et relation) d'un schéma car elles s'influencent toutes entre elles. Nous essayerons dans le chapitre quatre d'étudier ce point plus en détail.

Il est indiqué dans la présentation du modèle que les relations topologiques peuvent exister entre objets géo-graphiques, classes, couches, sous-couches mais cette fois il est précisé qu'il ne peut y avoir des cardinalités que pour les relations entre objets. Reprenons les propos exactes de l'auteur :

"R.151. L'application des relations topologiques entre deux classes, couches, sous-couches, ou toute combinaison entre elles ou avec un ou plusieurs objets, indique que les classe(s), couche(s) ou sous-couche(s) sont considérées comme des OG dont les types dépendent des types et du nombre d'objets qu'elles contiennent. Les règles appliquées aux OGS, OGC et OGX sont également applicables à de tels types de relations, y compris la possibilité d'orienter l'application d'une relation.

Cette règle est illustrée par un exemple repris ci-dessous :





"La superposition totale orientée (St) entre les classes "MAISONS" et "PARCELLES" signifie que chaque instance "Maison" est superposée totalement à une ou plusieurs instances "Parcelle", tandis que (Sp) orientée traduit soit que : a) chaque instance "Parcelle" est superposée partiellement par une ou plusieurs instances "Maison"; b) seulement certaines des instances "Parcelle" sont superposées partiellement par une ou plusieurs instances "Maison"; c) Seulement certaines des instances " parcelle" sont superposées totalement par une ou plusieurs instances "Maison"; d) nous sommes en présence d'une combinaison des cas mentionnés."

Ci-dessus nous avons repris les seuls renseignements présentés sur les relations topologiques entre couches , classes et sous -couches. Nous pouvons en déduire (notamment avec les définitions des propriétés topologiques entre des objets composés voir point 3.3.2) qu'une classe d'objets géo-graphiques est superposée à une autre classe d'objets géo-graphiques si au moins une instance d'objets de la première classe est superposée à une instance d'un objet de la deuxième classe. De même une classe d'objets géo-graphiques est voisine d'une classe d'objets géo-graphiques s'il existe au moins une instance d'un objet contenu dans la première classe qui est voisine d'une instance d'un objet de la deuxième classe. Avant de définir la relation topologique entre classes, nous avons décidé de compléter le cluster classe d'objets géo-graphiques par des opérations topologiques. Remarquons que toutes les contraintes topologiques ne peuvent être définies entre toutes les classes, elles dépendent des objets contenus dans la classe mais aussi des contraintes topologiques définies dans les classes. Nous ne définirons pas tous les cas suivant les contraintes topologiques contenues dans la classe.

CLUSTER CLASSE[OGS] *classe contenant un objet géo-graphique simple*
exports all
is CP[Alogique : AT_LOG, Vpermis : VOISPERMIS, Spermis : SUPPERMIS, Sobli : SUPOBLI, Vobli : VOISOBLI, Ens : SET[OGS]]
asserts
$\forall$ o : o $\in$ Ens(c) $\wedge$ Is-of-OGS_P(o) $\Rightarrow$ Vobli(c)= 'rien' $\wedge$ Vpermis(c)= 'rien' $\wedge$ Sobli(c) $\neq$ '+Spobli' $\wedge$ Sobli(c) $\neq$ '-Spobli' $\wedge$ (Spermis = 'Sn,t' $\vee$ Spermis = 'rien')
$\forall$ o : o $\in$ Ens(c) $\wedge$ Is-of-OGS_L(o) $\Rightarrow$ Vobli(c)= 'rien' $\wedge$ Vpermis(c)= 'rien'
Pour les invariants voir le cluster CLASSE[OGS] défini avant
hints
o, o' : OGS
c : CLASSE[OGS]

OPERATION +S(c, c') *superposition entre deux classes*
arity CLASSE[OGS] $\times$ CLASSE[OGS] $\rightarrow$ BOOLEAN
pré
post +S(c, c') $\Leftrightarrow$ $\exists$ o, o' : o $\in$ Ens(c) $\wedge$ o' $\in$ Ens(c') $\wedge$ +S(o, o')
hints
o, o' : OGS
OPERATION +St(c, c') *superposition totale entre deux classes*
arity CLASSE[OGS] $\times$ CLASSE[OGS] $\rightarrow$ BOOLEAN
pré $\neg$(((Is-of-OGS_L(o) $\vee$ Is-of-OGS_POL(o)) $\wedge$ Is-of-OGS_P(o')) $\vee$ (Is-of-OGS_POL(o) $\wedge$ Is-of-OGS_L(o'))) $\wedge$ o $\in$ Ens(c) $\wedge$ o' $\in$ Ens(c')
post +St(c, c') $\Leftrightarrow$ $\forall$o, p $\exists$ p', o' : o $\in$ Ens(c) $\wedge$ p $\in$ Représentation(o) $\wedge$ o' $\in$ Ens(c') $\wedge$ p' $\in$ Représentation(c') $\wedge$ Sup(p, p') $\wedge$ +S(c, c')
hints
p, p' : POINT
o, o' : OGS
OPERATION +Sp(c, c') *superposition partielle entre deux classes*
arity CLASSE[OGS] $\times$ CLASSE[OGS] $\rightarrow$ BOOLEAN
pré $\neg$(Is-of-OGS_P(o)) $\wedge$ o $\in$ Ens(c)
post (((Is-of-OGS_L(o) $\vee$ Is-of-OGS_POL(o)) $\wedge$ Is-of-OGS_P(o')) $\vee$ (Is-of-OGS_POL(o) $\wedge$ Is-of-OGS_L(o'))) $\wedge$ o $\in$ Ens(c) $\wedge$ o' $\in$ Ens(c')
$\Rightarrow$ +Sp(c, c') $\Leftrightarrow$ +S(c, c')
(Is-of-OGS_L(o) $\vee$ Is-of-OGS_POL(o)) $\wedge$ Is-of-OGS_POL(o')
$\Rightarrow$ +Sp(c, c') $\Leftrightarrow$ +S(c, c') $\wedge$ $\neg$(+St(c, c'))
hints
o, o' : OGS
OPERATION +Sn(c, c') *superposition nulle entre deux classes*
arity CLASSE[OGS] $\times$ CLASSE[OGS] $\rightarrow$ BOOLEAN
pré
post +Sn(c, c') $\Leftrightarrow$ $\neg$(+S(o, o'))
hints

o, o' : OGS
OPERATION +V(c, c') *voisinage entre deux classes*
arity CLASSE[OGS] × CLASSE[OGS] → BOOLEAN
pré (Is-of-OGS_POL(o) ∨ Is-of-OGS_POL(o')) ∧ o ∈ Ens(c) ∧ o' ∈ Ens(c')
post +V(c, c') ⇔ ∃ o, o' : o ∈ Ens(c) ∧ o' ∈ Ens(c') ∧ +V(o, o')
hints
o, o' : OGS
OPERATION +Vt(c, c') *voisinage total entre deux classes*
arity CLASSE[OGS] × CLASSE[OGS] → BOOLEAN
pré (Is-of-OGS_POL(o) ∨ Is-of-OGS_POL(o')) ∧ ¬(Is-of-OGS_P(o')) ∧ o ∈ Ens(c) ∧ o' ∈ Ens(c')
post
(Is-of-OGS_POL(o) ∧ Is-of-OGS_L(o'))
⇒ +Vt(c, c') ⇔ ∀o, p ∃ p', o' : o ∈ Ens(c) ∧ p ∈ Frontière(Représentation(o)) ∧ o' ∈ Ens(c') ∧ p' ∈(Représentation(o')) ∧ +S(p, p')
((Is-of-OGS_P(o) ∨ Is-of-OGS_L(o)) ∧ Is-of-OGS_POL(o'))
⇒ +Vt(c, c') ⇔ ∀o, p ∃ p', o' : o ∈ Ens(c) ∧ p ∈ Représentation(o) ∧ o' ∈ Ens(c') ∧ p' ∈ Frontière(Représentation(o')) ∧ +S(p, p')
(Is-of-OGS_POL(o) ∧ Is-of-OGS_POL(o'))
⇒ +Vt(c, c') ⇔ ∀o, p ∃ p', o' : o ∈ Ens(c) ∧ p ∈ Frontière(Représentation(o)) ∧ o' ∈ Ens(c') ∧ p' ∈ Frontière(Représentation(o')) ∧ +S(p, p')
hints
o, o' : OGS
OPERATION +Vp(c, c') *voisinage partiel entre deux classes*
arity CLASSE[OGS] × CLASSE[OGS] → BOOLEAN
pré (Is-of-OGS_POL(o) ∨ Is-of-OGS_POL(o')) ∧ ¬(Is-of-OGS_P(o)) ∧ o ∈ Ens(c) ∧ o' ∈ Ens(c')
post
((Is-of-OGS_POL(o) ∧ Is-of-OGS_L(o')) ∨ (Is-of-OGS_L(o) ∧ Is-of-OGS_POL(o')) ∨ (Is-of-OGS_POL(o) ∧ Is-of-OGS_POL(o'))) ⇒+Vp(c, c') ⇔ ¬(+Vt(c, c')) ∧ +V(c, c')
(Is-of-OGS_P(o) ∧ Is-of-OGS_POL(o'))⇒ +Vp(c, c') ⇔ +V(c, c')
hints
o, o' : OGS
OPERATION +Vn(c, c') *voisinage nul entre deux classes*
arity CLASSE[OGS] × CLASSE[OGS] → BOOLEAN
pré (Is-of-OGS_POL(o) ∨ Is-of-OGS_POL(o')) ∧ o ∈ Ens(c) ∧ o' ∈ Ens(c')
post +Vn(c, c') ⇔ ¬(+V(c, c'))
hints : o, o' : OGS
END CLASSE[OGS]

Nous devons maintenant définir les opérations pour une classe contenant deux objets puis trois...mais pour ne pas surcharger ce travail nous passons directement à la définition du cluster de relation topologique entre classes contenant un OGS..

CLUSTER REL_TOPO[CLASSE[OGS], CLASSE[OGS]] *relation topologique entre deux classes contenant chacun un objet géo-graphiques simples*
exports all

is CP[Srel1 : SUPREL1, Vrel1 : VOISREL1,Srel1 : SUPREL2, Vrel2 : VOISREL2, Rel : CP[CLASSE[OGS], CLASSE[OGS']]]

asserts

Sel1(Rel(r)) ≠ Sel2(Rel(r))
∀ o, o' : Is-of-OGS_P(o) ∧ Is-of-OGS_P(o') ∧ o ∈ Ens(c) ∧ o' ∈ Ens(c')⇒ Vrel1(r)= 'rien'∧ Vrel2(r)= 'rien'
∀ o, o' : Is-of-OGS_L(o) ∧ Is-of-OGS_L(o') ∧ o ∈ Ens(c) ∧ o' ∈ Ens(c')⇒ Vrel1(r) = 'rien' ∧ Vrel2(r)= 'rien'
∀ o, o' : Is-of-OGS_P(o) ∧ Is-of-OGS_L(o') ∧ o ∈ Ens(c) ∧ o' ∈ Ens(c') ⇒ Vrel1(r)= 'rien' ∧ Vrel2(r)= 'rien' ∧ Srel1(r) ≠ '←+St' ∧ Srel1(r) ≠ '←-St'
∀ o, o' : Is-of-OGS_L(o) ∧ Is-of-OGS_P(o') ∧ o ∈ Ens(c) ∧ o' ∈ Ens(c') ⇒ Vrel1(r)= 'rien' ∧ Vrel2(r)= 'rien' ∧ Srel2(r) ≠ '+St→' ∧ Srel2(r) ≠ '-St→'
∀ o, o' : Is-of-OGS_P(o) ∧ Is-of-OGS_POL(o') ∧ o ∈ Ens(c) ∧ o' ∈ Ens(c') ⇒ Vrel1(r) ≠ '←+Vt' ∧ Vrel1(r) ≠ '←-Vt' ∧ Srel1(r) ≠ '←+St' ∧ Srel1(r) ≠ '←-St'
∀ o, o' : Is-of-OGS_POL(o) ∧ Is-of-OGS_P(o') ∧ o ∈ Ens(c) ∧ o' ∈ Ens(c') ⇒ Vrel2(r) ≠ '+Vt→' ∧ Vrel2(r) ≠ '-Vt→' ∧ Srel2(r) ≠ '+St→' ∧ Srel2(r) ≠ '-St→'
∀ o, o' : Is-of-OGS_L(o) ∧ Is-of-OGS_POL(o') ∧ o ∈ Ens(c) ∧ o' ∈ Ens(c') ⇒ Srel1(r) ≠ '←+St' ∧ Srel1(r) ≠ '←-St'
∀ o, o' : Is-of-OGS_POL(o) ∧ Is-of-OGS_L(o') ∧ o ∈ Ens(c) ∧ o' ∈ Ens(c') ⇒ Srel2(r) ≠ '+St→' ∧ Srel2(r) ≠ '-St→'

Vrel(r) = '+Vn' ⇒ (<c, c'> = Rel(r) ⇔ +Vn(c, c'))
Vrel(r) = '-Vn' ⇒ (<c, c'> = Rel(r) ⇔ -Vn(c, c'))
Vrel(r) = '+Vt→' ⇒ (<c, c'>, Rel(r)) ⇔ +Vt(c, c'))
Vrel(r) = '←+Vt' ⇒ (<c, c'> = Rel(r)) ⇔ +Vt(c', c))
Vrel(r) = '-Vt→' ⇒ (<c, c'> = Rel(r)) ⇔ -Vt(c, c'))
Vrel(r) = '←-Vt' ⇒ (<c, c'> = Rel(r)) ⇔ -Vt(c', c))
Vrel(r) = '+Vp→' ⇒ (<c, c'> = Rel(r)) ⇔ +Vp(c, c'))
Vrel(r) = '←+Vp' ⇒ (<c, c'> = Rel(r)) ⇔ +Vp(c', c))
Vrel(r) = '-Vp→' ⇒ (<c, c'> = Rel(r)) ⇔ -Vp(c, c'))
Vrel(r) = '←-Vp' ⇒ (<c, c'> = Rel(r)) ⇔ -Vp(c', c))
Srel(r) = '+Sn' ⇒ (<c, c'> = Rel(r)) ⇔ +Sn(c, c'))
Srel(r) = '-Sn' ⇒ (<c, c'> = Rel(r)) ⇔ -Sn(c, c'))
Srel(r) = '+St→' ⇒ (<c, c'> = Rel(r)) ⇔ +St(c, c'))
Srel(r) = '←+St' ⇒ (<c, c'> = Rel(r)) ⇔ +St(c', c))
Srel(r) = '-St→' ⇒ (<c, c'> = Rel(r)) ⇔ -St(c, c'))
Srel(r) = '←-St' ⇒ (<c, c'> = Rel(r)) ⇔ -St(c', c))
Srel(r) = '+Sp→' ⇒ (<c, c'> = Rel(r)) ⇔ +Sp(c, c'))
Srel(r) = '←+Sp' ⇒ (<c, c'> = Rel(r)) ⇔ +Sp(c', c))
Srel(r) = '-Sp→' ⇒ (<c, c'> = Rel(r)) ⇔ -Sp(c, c'))
Srel(r) = '←-Sp' ⇒ (<c, c'> = Rel(r)) ⇔ -Sp(c', c))

hints

o, o' : OGS
c, c' : CLASSE[OGS]
r : REL_TOPO[CLASSE[OGS], CLASSE[OGS]]

END REL_TOPO[CLASSE, CLASSE]

3.3.1.5. Les couches et les sous-couches

Les couches et les sous- couches sont des groupements de classes et de relations auxquelles peuvent être ajoutées des contraintes topologiques obligatoires et permises comme pour les classes d'objets géo-graphiques. Ces contraintes portent sur toutes les instances des objets géo-graphiques contenus dans les classes de la couche ou sous-couche.

Nous ne détaillerons pas ces deux concepts, nous nous contentons de donner l'expression de type très général.

COUCHE[CLASSE, RELATION, CLASSE, SOUS-COUCHE,...]
is CP[CP[CLASSE, RELATION, CLASSE, SOUS-COUCHE,...],VOISOBLI, SUPOBLI, SUPPERMIS, VOISPERMIS]
SOUS-COUCHE[CLASSE, RELATION, CLASSE, RELATION,...]
is CP[CP[CLASSE, RELATION, CLASSE, RELATION,...],VOISOBLI, SUPOBLI, SUPPERMIS, VOISPERMIS]
RELATION is UNION[REL_TOPO[OGS, OGS], REL_TOPO[CLASSE, CLASSE], REL_LOG[OBJET, OBJET], REL_LOG[CLASSE, CLASSE], ...]

Nous avons vu que les relations logiques peuvent exister entre les couches, les sous-couches, les classes, les objets et leurs combinaisons. Pour définir formellement les relations logiques en tenant compte des couches et des sous-couches nous pouvons remplacer le paramètre classe de la définition de la relation logique vue plus haut par un paramètre formel plus général groupant classe, sous-couche et couche :

AGREGAT is UNION[CLASSE, COUCHE, SOUS-COUCHE]

Pour spécifier les relations topologiques entre couches et entre sous-couches nous pouvons définir les opérations topologiques de ces clusters comme nous l'avons fait pour le cluster objet et le cluster classe. Ceci permettra de définir facilement les relations topologiques entre couches et entre sous-couches. Il y a superposition de deux couches s'il y a superposition d'une instance d'un objet géo-graphique d'une classe contenue dans la première couche et d'une instance d'un objet d'une classe de la deuxième couche. etc...

3.3.2 Extension de la formalisation aux objets géo-graphiques composés et complexes.

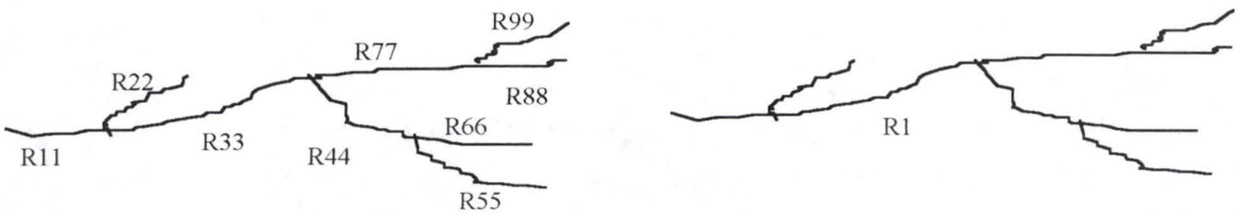
Nous allons maintenant essayer d'étendre nos définitions formelles en considérant les objets géo-graphiques composés (OGC) et les objets géo-graphiques complexes (OGX). Pour définir exactement ce qu'on entend par les OGC reprenons une partie des propos de l'auteur sur les OGC.

"Les objets géo-graphiques composés sont des objets qui proviennent de la réunion ou de la division des objets simples du même type, appartenant à la même classe ou à des classes différentes."[...]

"Exemple : l'Etat X est composé de communes (chaque commune appartient à la classe communes), alors l'Etat x constitue un OGC du type surface homogène."[...]

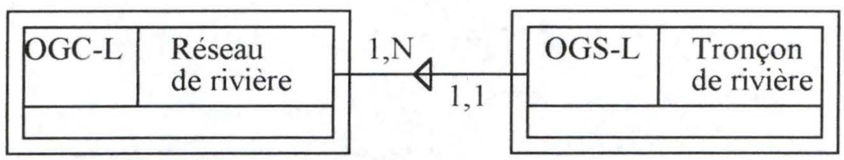
"R.14. Les objets géo-graphiques entrant dans la composition d'objets géo-graphiques composés ne gardent pas nécessairement leur propre identité."

Un exemple de représentation d'objets géo-graphiques composés avec et sans l'identité des composants est donné :

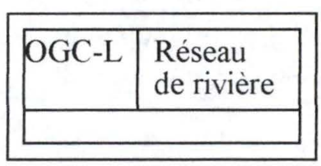


"OGC : R1=R11+R22+R33+R44+R55+R66+R77+R88+R99. Si on supprime les identités des composants de R1, on obtient un OGC dont les composants n'ont pas leur propre identité. Par exemple un cours d'eau dont on n'a pas identifié les tronçons composants."

La modélisation du dessin de gauche de ci-dessus suivant les règles du modèle CONGOO donne :



Par contre pour le dessin de droite nous avons[1]



L'objet géo-graphique composé semble signifier qu'il a une représentation composée de plusieurs éléments géométriques (point, ligne, polygone) de même type. Nous avons donc décidé de définir les objets géo-graphiques composés comme des objets géo-graphiques qui ont une représentation composée c'est-à-dire composée de points ou de lignes ou de polygones.

La représentation d'un objet composé peut être divisée en trois types : le type composé point, le type composé ligne, le type composé polygone. Trois clusters peuvent donc être définis.

[1] Notons que l'auteur précise qu'une relation composé-composant peut s'appliquer à un OGC dont les composants n'ont pas leur propre identité pour faciliter l'identification et la structuration des éléments à numériser.

CLUSTER COMP_POINT *représentation composée de points*
exports all
is SET[POINT]
asserts
$Card(cp) \geq 2$
hints
cp : COMP_POINT

END COMP_POINT

CLUSTER COMP_LIGNE *représentation composée de lignes*
exports all
is SET[LIGNE]
asserts
$Card(cl) \geq 2$
hints
cl : COMP_LIGNE

OPERATION $\in$ *un point est membre de la représentation comp-ligne*
arity POINT $\times$ COMP_LIGNE $\rightarrow$ BOOLEAN
pré
post $p \in cl \Rightarrow \exists\, l : l \in cl \wedge p \in l$
hints
cl : COMP_LIGNE
l : LIGNE
p : POINT
END COMP_LIGNE

CLUSTER COMP_POLYGONE *représentation composée de polygone*
exports all
is SET[POLYGONE]
asserts
$Card(cpol) \geq 2$
hints
cpol : COMP_POLYGONE

OPERATION Frontière
arity COMP_POLYGONE $\rightarrow$ COMP_LIGNE
pré
post $\forall l : l \in cl \Rightarrow \exists\, pol : pol \in cpol \wedge l = Frontière(pol)$
$\wedge\ Card(cpol) = Card(cl)$
hints
pol : POLYGONE
cpol : COMP_POLYGONE
l : LIGNE
cl : COMP_LIGNE

```
OPERATION Limite *Limite externe*
 arity COMP_POLYGONE → COMP_POINT
 pré
post ∀p : p ∈ cp ⇒ ∃ i,l : 1 ≤ i ≤ Length(l) ∧ l ∈ Limite(cpol) ∧ Si(l, i) = p ∧
(∀ pol : pol ∈ cpol ⇒ ¬(Sup(p, pol)))
∧ ∀p,pol Sup(p, pol) ⇒ ∃ p' : p' ∈ Intérieur(pol) ∧ Egal(p,p')
 hints
i: INTEGER
p, p' : POINT
pol : POLYGONE
cpol : COMP_POLYGONE
l : LIGNE
cl : COMP_LIGNE
OPERATION ∈
 arity POINT × COMP_POLYGONE → BOOLEAN
 pré
post p ∈ cpol ⇒∃ pol ∈ cpol ∧ p ∈ pol
 hints
p : POINT
pol : POLYGONE
cpol : COMP_POLYGONE
END COMP_POLYGONE
```

De même que pour les objets géo-graphiques composés, nous avons choisi de définir les objets géo-graphiques complexes comme des objets ayant une représentation complexe. La représentation d'un objet géo-graphique de type complexe est divisée en quatre sous-catégories : celle composée d'un ensemble de points et de lignes, celle composée d'un ensemble de points et de polygones, celle composée d'un ensemble de lignes et de polygones et celle composée d'un ensemble de points, de lignes et de polygones.

```
CLUSTER COMB_PT_L *représentation complexe composée de points et lignes*
 exports all
 is CP[ Ensp : SET[POINT], Ensl : SET[LIGNE]]
 asserts
*les deux ensembles ne sont pas vides*
¬(Empty?(Ensp(co))) ∧ ¬((Empty?(Ensl(co)))
 hints
co : COMB_PT_L
-------------------------------------------------------------
OPERATION ∈
 arity POINT × COMB_PT_L → BOOLEAN
 pré
post p ∈ co ⇒∃ l : l ∈ Ensl(co) ∧ p ∈ l ∨ p ∈ Ensp(co)
 hints
co : COMB_PT_L
l : LIGNE
p : POINT
END COMB_PT_L
```

CLUSTER COMB_L_POL *représentation complexe composée de lignes et de polygones*
exports all
is CP[Ensl : SET[LIGNE], Enspol : SET[POLYGONE]]
asserts
les deux ensembles ne sont pas vides
$\neg$(Empty?(Ensl(co))) $\wedge$ $\neg$((Empty?(Enspol(co)))
hints
co : COMB_L_POL

- -

OPERATION $\in$
arity POINT $\times$ COMB_L_POL $\rightarrow$ BOOLEAN
pré
post p $\in$ co $\Rightarrow \exists$ l : l $\in$ Ensl(co) $\wedge$ p $\in$ l $\vee$ $\exists$ pol : pol $\in$ Enspol(co) $\wedge$ p $\in$ pol
hints
co : COMB_L_POL
l : LIGNE
p : POINT
pol : POLYGONE
OPERATION Frontière
arity COMB_L_POL $\rightarrow$ COMP_LIGNE
pré
post
$\forall$l : l $\in$ cl $\Rightarrow$ $\exists$ pol : pol $\in$ Enspol(co) $\wedge$ l = Frontière(pol) $\wedge$ Card(Enspol((co)) = Card(cl)
hints
pol : POLYGONE
co : COMB_L_POL
l : LIGNE
cl : COMP_LIGNE
END COMB_L_POL

CLUSTER COMB_PT_POL*représentation complexe composée de points et de polygones*
exports all
is CP[Ensp : SET[POINT], Enspol : SET[POLYGONE]]
asserts
les deux ensembles ne sont pas vides
$\neg$(Empty?(Ensp(co))) $\wedge$ $\neg$((Empty?(Enspol(co)))
hints
co : COMB_PT_POL

- -

OPERATION $\in$
arity POINT $\times$ COMB_PT_POL $\rightarrow$ BOOLEAN
pré
post p $\in$ co $\Rightarrow$ p $\in$ Ensp(co) $\vee$ $\exists$ pol : pol $\in$ Enspol(co) $\wedge$ p $\in$ pol
hints
co : COMB_PT_POL
p : POINT
pol : POLYGONE
OPERATION Frontière

arity COMB_PT_POL → COMP_LIGNE
pré
post ∀l : l ∈ cl ⇒ ∃ pol : pol ∈ Enspol(co) ∧ l = Frontière(pol)
∧ Card(Enspol((co)) = Card(cl)
hints
pol : POLYGONE
co : COMB_PT_POL
l : LIGNE
cl : COMP_LIGNE
END COMB_PT_POL

CLUSTER COMB_PT_L_POL *représentation complexe composée de points, de lignes et de polygones*
exports all
is CP[Ensp : SET[POINT], Ensl : SET[LIGNE], Enspol : SET[POLYGONE]]
asserts
les trois ensembles ne sont pas vides
¬(Empty?(Ensp(co))) ∧ ¬((Empty?(Ensl(co))) ∧ ¬((Empty?(Enspol(co)))
hints
co : COMB_PT_L_POL

OPERATION ∈
arity POINT × COMB_PT_L_POL → BOOLEAN
pré
post p ∈ co ⇒p ∈ Ensp(co) ∨ ∃ l : l ∈ Ensl(co) ∧ p ∈ l ∨ ∃ pol : pol ∈ Enspol(co) ∧ p ∈ pol
hints
co : COMB_PT_L_POL
l : LIGNE
p : POINT
pol : POLYGONE
OPERATION Frontière
arity COMB_PT_L_POL → COMP_LIGNE
pré
post ∀l : l ∈ cl ⇒ ∃ pol : pol ∈ Enspol(co) ∧ l = Frontière(pol)
∧ Card(Enspol((co)) = Card(cl)
hints
pol : POLYGONE
co : COMB_PT_L_POL
l : LIGNE
cl : COMP_LIGNE
END COMB_PT_L_POL

A présent nous pouvons définir le cluster représentation en tenant compte de toutes les représentations mais quelques remarques doivent être effectuées auparavant.

Tout d'abord, dans le modèle, il est donné les relations topologiques possibles entre chaque représentation d'OGS (voir chapitre 2 Figure 2.9), entre chaque représentation d'OGC et entre chaque représentation d'OGX, mais il n'y a pas d'information sur les relations topologiques qui peuvent exister entre les représentations d'OGS et les représentations d'OGC ou entre les représentations d'OGS et les représentations d'OGX ou entre les représentations d'OGC et les représentations d'OGX. Nous avons choisi de spécifier dans la suite quand une relation topologique peut exister pour ces combinaisons en gardant la philosophie de l'auteur.

Ensuite, il faut signaler que la limite de la représentation comp_polygone ou la limite d'une représentation complexe (composée de polygones) utilisée pour identifier la propriété de voisinage est égale à la somme de toutes les limites des polygones "contenus" dans ces représentations.

Enfin signalons qu'une instance de type point peut être à la fois superposée et voisine à une instance d'une représentation composée ou complexe. Par exemple, si une représentation complexe est composée d'un polygone et d'une ligne passant par la limite de ce polygone et qu'une représentation de type point est située sur la limite du polygone et sur la ligne de la représentation complexe, la représentation de type point est considérée voisine et superposée à la représentation complexe.

CLUSTER REPRESENTATION
exports all
is UNION[POINT, LIGNE, POLYGONE, COMP_POINT, COMP_LIGNE, COMP_POLYGONE, COMB_PT_L, COMB_L_POL, COMB_PT_POL, COMB_PT_L_POL, ALPHA]
asserts
hints

- -

OPERATION Dim *donne la dimension d'une représentation*
arity REPRESENTATION → INTEGER
pré ¬(Is-of-ALPHA(r))
post
Dim(r) = i ∧
(Is-of-POINT(r) ∨ Is-of-COMP_POINT(r))⇒ i = 0
(Is-of-LIGNE(r) ∨ Is-of-COMP_LIGNE(r) ∨ Is-of-COMB_P_L(r))⇒ i = 1
(Is-of-POLYGONE(r) ∨ Is-of-COMP_POLYGONE(x) ∨ Is-of-COMB_P_POL(x) ∨ Is-of-COMB_L_POL(r) ∨ Is-of-_P_L_POL(r))⇒ i = 2)
hints
r : REPRESENTATION
i : INTEGER
OPERATION S(r, r') *superposition entre deux représentations*
arity REPRESENTATION × REPRESENTATION → BOOLEAN
pré ¬(Is-of-ALPHA(r)) ∧ ¬(Is-of-ALPHA(r'))
post
(Dim(r) = 0 ∨ Dim(r) = 1)∧ (Dim(r') = 0 ∨ Dim(r') = 1)
⇒ S(r, r') ≡ ∃ x,y : x ∈ r ∧ y ∈ r' ∧ Sup(x, y)

(Dim(r) = 0 ∨ Dim(r) = 1) ∧ (Dim(r') = 2)

$\Rightarrow S(r, r') \equiv \exists x,y : x \in r \wedge y \in$ intérieur$(r') \wedge Sup(x, y)$

$(Dim(r) = 2) \wedge (Dim(r) = 0 \vee Dim(r) = 1)$
$\Rightarrow S(r, r') \equiv \exists x,y : x \in Interieur(r) \wedge y \in r' \wedge Sup(x, y)$

$(Dim(r) = 2) \wedge (Dim(r') = 2)$
$\Rightarrow S(r, r') \equiv \exists x,y : x \in r \wedge y \in r' \wedge Sup(x, y) \wedge \neg (x \in$ Frontière$(r) \wedge y \in$ Frontière$(r'))$
hints
r, r' : REPRESENTATION
x,y : POINT
OPERATION +Sn(r, r') *superposition nulle entre deux représentations*
arity REPRESENTATION $\times$ REPRESENTATION $\rightarrow$ BOOLEAN
pré $\neg$(Is-of-ALPHA(r)) $\wedge$ $\neg$(Is-of-ALPHA(r'))
post +Sn(r, r') $\equiv$ $\neg$(S(r, r'))
hints
r, r': REPRESENTATION
OPERATION +St(r, r') *superposition totale entre deux représentations*
arity REPRESENTATION $\times$ REPRESENTATION $\rightarrow$ BOOLEAN
pré $\neg$(Is-of-ALPHA(r)) $\wedge$ $\neg$(Is-of-ALPHA(r')) $\wedge$
$\neg((Dim(r) = 1 \vee Dim(r) = 2) \wedge Dim(r') = 0) \wedge$
$\neg((Dim(r) = 2) \wedge (Dim(r') = 1) \wedge \neg$(Is-of-COMP_POINT(r) $\wedge$ Is-of-POINT(r'))
post
$(Dim(r) = 0 \vee Dim(r) = 1) \wedge (Dim(r') = 1)$
$\Rightarrow +St(r, r') \equiv \forall x \exists y : x \in r \wedge y \in r' \wedge Sup(x, y)$

$(Dim(r) = 0 \vee Dim(r) = 1) \wedge (Dim(r') = 2)$
$\Rightarrow +St(r, r') \equiv \forall x \exists y : x \in r \wedge y \in r' \wedge Sup(x, y) \wedge S(r, r')$

Is-of-POINT(r) $\wedge$ Is-of-POINT(r') $\Rightarrow$ +St(r, r') $\equiv$ Sup(r,r')

Is-of-POINT(r) $\wedge$ Is-of-COMP_POINT(r') $\Rightarrow$ +St(r, r') $\equiv \exists y : y \in r' \wedge S(r, y)$

$(Dim(r) = 2) \wedge (Dim(r') = 2)$
$\Rightarrow +St(r, r') \equiv \forall x \exists y: x \in r \wedge y \in r' \wedge Sup(x, y)$
hints
r, r' : REPRESENTATION
x,y : POINT
OPERATION +Sp(r, r') *superposition partielle entre deux représentations*
arity REPRESENTATION $\times$ REPRESENTATION $\rightarrow$ BOOLEAN
pré $\neg$(Is-of-ALPHA(r)) $\wedge$ $\neg$(Is-of-ALPHA(r'))$\wedge$ $\neg$ (Is-of-POINT(r))
post
$((Dim(r) = 1 \vee Dim(r) = 2) \wedge Dim(r') = 0)$
$\vee (Dim(r) = 2 \wedge Dim(r') = 1) \vee$ (Is-of-COMP_POINT(r) $\wedge$ Is-of-POINT(r'))
$\Rightarrow$ +Sp(r, r') $\equiv$ S(r, r')

((Is-of-COMP_POINT(r) $\vee$ Dim(r) = 1 $\vee$ Dim(r) = 2) $\wedge$ Dim(r') = 2) $\vee$ ((Is-of-COMP_POINT(r) $\vee$ Dim(r) = 1) $\wedge$ Dim(r') = 1) $\vee$ (Is-of-COMP_POINT(r) $\wedge$ Is-of-COMP_POINT(r'))$\Rightarrow$ +Sp(r, r') $\equiv$ S(r, r') $\wedge$ $\neg$(+St(r, r'))

hints
r, r' : REPRESENTATION
OPERATION V(r, r') *voisinage entre deux représentations*
arity REPRESENTATION × REPRESENTATION → BOOLEAN
pré $\neg$(Is-of-ALPHA(r)) $\wedge$ $\neg$(Is-of-ALPHA(r')) $\wedge$
$\neg((\mathrm{Dim}(r) = 0 \vee \mathrm{Dim}(r) = 1) \wedge (\mathrm{Dim}(r') = 0 \vee \mathrm{Dim}(r') = 1)$
post
$((\mathrm{Dim}(r) = 0 \vee \mathrm{Dim}(r) = 1) \wedge \mathrm{Dim}(r') = 2)$
$\Rightarrow V(r, r') \equiv \exists x, y : x \in r \wedge y \in \text{Frontière}(r') \wedge \text{Sup}(x, y)$
$\equiv S(r, \text{Frontière}(r'))$
$(\mathrm{Dim}(r) = 2 \wedge (\mathrm{Dim}(r') = 0 \vee \mathrm{Dim}(r') = 1))$
$\Rightarrow V(r, r') \equiv S(\text{Frontière}(r), r')$

$(\mathrm{Dim}(r) = 2 \wedge \mathrm{Dim}(r') = 2)$
$\Rightarrow V(r, r') \equiv S(\text{Frontière}(r), \text{Frontière}(r'))$
hints
x, y : POINT
r, r' : REPRESENTATION
OPERATION +Vt(r, r') *voisinage total entre deux représentations*
arity REPRESENTATION × REPRESENTATION → BOOLEAN
pré $\neg$(Is-of-ALPHA(r)) $\wedge$ $\neg$(Is-of-ALPHA(r')) $\wedge$
$\neg((\mathrm{Dim}(r) = 0 \vee \mathrm{Dim}(r) = 1) \wedge (\mathrm{Dim}(r') = 0 \vee \mathrm{Dim}(r') = 1)) \wedge$
$\neg(\mathrm{Dim}(r') = 0)$
post
$((\mathrm{Dim}(r) = 0 \vee \mathrm{Dim}(r) = 1) \wedge \mathrm{Dim}(r') = 2)$
$\Rightarrow +Vt(r, r') \equiv +St(r, \text{Frontière}(r'))$

$(\mathrm{Dim}(r) = 2 \wedge \mathrm{Dim}(r') = 1)$
$\Rightarrow +Vt(r, r') \equiv +St(\text{Frontière}(r), r')$

$(\mathrm{Dim}(r) = 2 \wedge \mathrm{Dim}(r') = 2)$
$\Rightarrow +Vt(r, r') \equiv +St(\text{Frontière}(r), \text{Frontière}(r'))$
hints
r, r' : REPRESENTATION
OPERATION +Vn(r, r') *voisinage nul entre deux représentations*
arity REPRESENTATION × REPRESENTATION → BOOLEAN
pré $\neg$(Is-of-ALPHA(r)) $\wedge$ $\neg$(Is-of-ALPHA(r')) $\wedge$
$\neg((\mathrm{Dim}(r) = 0 \vee \mathrm{Dim}(r) = 1) \wedge (\mathrm{Dim}(r') = 0 \vee \mathrm{Dim}(r') = 1))$
post $+Vn(r, r') \equiv \neg(V(r, r'))$
hints
r, r' : REPRESENTATION
OPERATION +Vp(r, r') *voisinage partiel entre deux représentations*
arity REPRESENTATION × REPRESENTATION → BOOLEAN
pré $\neg$(Is-of-ALPHA(r)) $\vee$ $\neg$(Is-of-ALPHA(r')) $\wedge$ $\neg$(Is-of-POINT(r)) $\wedge$
$\neg((\mathrm{Dim}(r) = 0 \vee \mathrm{Dim}(r) = 1) \wedge (\mathrm{Dim}(r') = 0 \vee \mathrm{Dim}(r') = 1))$
post
$(\mathrm{Dim}(r) = 2 \wedge (\mathrm{Dim}(r') = 0) \vee (\text{Is-of-COMP\_POINT}(r) = 2 \wedge (\mathrm{Dim}(r') = 2)$
$\Rightarrow +Vp(r, r') \equiv V(r, r')$

```
(Dim(r) = 1 ∧ Dim(r') = 2) ∨ (Dim(r) = 2 ∧ Dim(r') = 1) ∨
(Dim(r) = 2∧Dim(r') = 2)
⇒ +Vp(r, r') ≡ V(r, r') ∧ ¬(+Vt(r, r'))
 hints r, r' : REPRESENTATION
END REPRESENTATION
```

Nous pouvons à présent aussi définir les objets géo-graphiques composés et les objets géo-graphiques complexes.

```
CLUSTER OGC_P *objet géo-graphique composé du type point*
 exports all
 is OBJET
 asserts
*un objet géo-graphique composé de type point a une représentation de type comp_point*
Is-of-COMP_POINT[Représentation(o)]
 hints  o : OGC_P
------------------------------------------------------------------
END OGC_P
```

```
CLUSTER OGC_L *objet géo-graphique composé du type ligne*
 exports all
 is OBJET
 asserts
*un objet géo-graphique composé de type ligne a une représentation de type comp_ligne*
Is-of-COMP_LIGNE[Représentation(o)]
 hints  o : OGC_L
------------------------------------------------------------------
END OGC_L
```

```
CLUSTER OGC_POL *objet géo-graphique composé du type polygone*
 exports all
 is OBJET
 asserts
*un objet géo-graphique composé de type polygone a une représentation de type
comp_polygone*
Is-of-COMP_POLYGONE[Représentation(o)]
 hints  o : OGC_POL
------------------------------------------------------------------
END OGC_POL
```

```
CLUSTER OGX_P_L*objet géo-graphique complexe*
 exports all
 is OBJET
 asserts
*un objet géo-graphique complexe point-ligne a une représentation de type comb_p_l*
```

```
Is-of-COMB_P_L[Représentation(o)]
 hints o : OGX_P_L
------------------------------------------------------------------
END OGX_P_L
CLUSTER OGX_P_POL *objet géo-graphique complexe*
 exports all
 is OBJET
 asserts
*un objet géo-graphique complexe point-polygone a une représentation de type comb_p_pol*
Is-of-COMB_P_POL[Représentation(o)]
 hints o : OGX_P_POL
------------------------------------------------------------------
END OGX_P_POL
```

```
CLUSTER OGX_L_POL*objet géo-graphique complexe*
 exports all
 is OBJET
 asserts
*un objet géo-graphique complexe ligne-polygone a une représentation comb_l_pol*
Is-of-COMB_L_POL[Représentation(o)]
 hints o : OGX_L_POL
------------------------------------------------------------------
END OGX_L_POL
```

```
CLUSTER OGX_P_L_POL *objet géo-graphique complexe*
 exports all
 is OBJET
 asserts
*un objet géo-graphique complexe point-ligne-polygone a une représentation de type
comb_p_l_pol*
Is-of-COMB_P_L_POL[Représentation(o)]
 hints o : OGX_P_L_POL
------------------------------------------------------------------
END OGX_P_L_POL
```

Dans le modèle on parle souvent d'objets géo-graphiques simples (OGS), d'objets géo-graphiques composés (OGC) et d'objets géo-graphiques complexes (OGX) que nous pouvons définir par :

OGS is UNION[OGS_P, OGS_L, OGS_POL]
OGC is UNION[OGC_P, OGC_L, OGC_POL]
OGX is UNION[OGX_P_L, OGX_P_POL, OGX_L_POL, OGX_P_L_POL]

Nous pouvons aussi définir un type reprenant tous les types d'objets géo-graphiques.

```
CLUSTER OBGEO *objet géo-graphique *
 exports all
 is UNION[OGS, OGC, OGX]
 asserts
 hints
```

- -

OPERATION Dim *donne la dimension d'un objet géo-graphique*
arity OBGEO $\rightarrow$ INTEGER
pré
post
Dim(o) = Dim(Représentation(o))
hints
r : OBGEO
i : INTEGER
OPERATION +S(o, o') *superposition entre deux objets géo-graphiques*
arity OBGEO $\times$ OBGEO $\rightarrow$ BOOLEAN
pré
post +S(o, o') $\Leftrightarrow$ S(Représentation(o), Représentation(o'))
hints o, o' : OBGEO
OPERATION +Sn(o, o') *superposition nulle entre deux objets géo-graphiques*
arity OBGEO $\times$ OBGEO $\rightarrow$ BOOLEAN
pré
post +Sn(o, o') $\Leftrightarrow$ +Sn(Représentation(o), Représentation(o'))
hints o, o' : OBGEO
OPERATION +St(o, o') *superposition totale entre deux objets géo-graphiques*
arity OBGEO $\times$ OBGEO $\rightarrow$ BOOLEAN
pré $\neg$((Dim(o) = 1 $\vee$ Dim(o) = 2) $\wedge$ Dim(o') = 0) $\wedge$
$\neg$((Dim(o) = 2) $\wedge$ (Dim(o') = 1) $\wedge$ $\neg$(Is-of-OGC_P(o) $\wedge$ Is-of-OGS_P(o'))
post +St(o, o') $\Leftrightarrow$ +St(Représentation(o), Représentation(o'))
hints o, o' : OBGEO
OPERATION +Sp(o, o') *superposition partielle entre deux objets géo-graphiques*
arity OBGEO $\times$ OBGEO $\rightarrow$ BOOLEAN
pré $\neg$(Is-of-OGS_P(o))
post +Sp(o, o') $\Leftrightarrow$ +Sp(Représentation(o), Représentation(o'))
hints o, o' : OBGEO
OPERATION +V(o, o') *voisinage entre deux objets géo-graphiques*
arity OBGEO $\times$ OBGEO $\rightarrow$ BOOLEAN
pré $\neg$((Dim(o) = 0 $\vee$ Dim(o) = 1)$\wedge$ (Dim(o') = 0 $\vee$ Dim(o') = 1)
post +V(o, o') $\Leftrightarrow$ V(Représentation(o), Représentation(o'))
hints o, o' : OBGEO
OPERATION +Vt(o, o')*voisinage total entre deux objets géo-graphiques*
arity OBGEO $\times$ OBGEO $\rightarrow$ BOOLEAN
pré $\neg$((Dim(o) = 0 $\vee$ Dim(o) = 1)$\wedge$ (Dim(o') = 1)) $\wedge$
$\neg$(Dim(o') = 0)
post +Vt(o, o') $\Leftrightarrow$ +Vt(Représentation(o), Représentation(o'))
hints o, o' : OBGEO
OPERATION +Vp(o, o') *voisinage partiel entre deux objets géo-graphiques*
arity OBGEO $\times$ OBGEO $\rightarrow$ BOOLEAN
pré $\neg$(Is-of-OGS_P(o)) $\wedge$ $\neg$((Dim(o) = 0 $\vee$ Dim(o) = 1)$\wedge$ (Dim(o') = 0 $\vee$ Dim(o') = 1))
post +Vp(o, o') $\Leftrightarrow$ +Vp(Représentation(o), Représentation(o'))
hints o, o' : OBGEO
OPERATION +Vn(o, o') *voisinage nul entre deux objets géo-graphiques*
arity OBGEO $\times$ OBGEO $\rightarrow$ BOOLEAN

pré ¬((Dim(o) = 0 ∨ Dim(o) = 1)∧ (Dim(o') = 0 ∨ Dim(o') = 1))
post +Vn(o, o')⇔ ¬(+V(o, o'))
hints o, o' : OBGEO
* + les opérations -Vn, -Vt, -Vp, -Sn, -Sp, -Sp qui sont les négations des opérations ci-dessus*

END OBGEO

Les définitions des clusters d'une classe et d'une relation topologique en tenant compte des objets géo-graphiques composés et complexes sont définis d'une manière similaire à celles de la première partie. En ce qui concerne le cluster des relations logiques rien ne change.

Des contraintes peuvent être définies entre deux relations. Ces relations peuvent être des relations logiques ou topologiques ou encore des relations composé/composant ou de généralisation-spécialisation que nous n'avons pas étudiées ici. Un type "générique" reprenant les deux types de relations que nous avons étudiés a été défini pour ne pas multiplier les définitions. Nous définissons ci-dessous les contraintes pour des relations entre objets mais dans le modèle il est indiqué qu'elles peuvent être appliquées à des paires de relations entre classes d'objets, couches et sous-couches. Ces cas n'ont pas été étudiés dans ce travail.
RELATOB is UNION[REL_LOG[<u>OBJET</u>, <u>OBJET</u>], REL_TOPO[<u>OBJET</u>, <u>OBJET</u>]]

CLUSTER CONTRAINTE *contrainte entre deux relations*
exports all
- -
-
OPERATION Exclusion
arity <u>RELATOB</u> × <u>RELATOB</u> × SET[<u>OBJET</u>] → BOOLEAN
pré (Cens1(rel1) = ens ∨ Cens2(rel1) = ens) ∧ (Cens1(rel2) = ens ∨ Cens2(rel2) = ens)
post
Cens1(rel1) = ens ∧ Cens1(rel2) = ens ⇒ (Exclusion(rel1, rel2 , ens) ≡ ∀o : o ∈ ens ⇒ (
¬(Empty?(Filter(rel1, Sel1(rel1i = o)))) ⇒ Empty?(Filter(rel2, Sel1(rel2i = o))) ∧
¬(Empty?(Filter(rel2, Sel1(rel2i = o)))) ⇒ Empty?(Filter(rel1, Sel1(rel1i = o)))))
Cens1(rel1) = ens ∧ Cens2(rel2) = ens ⇒ (Exclusion(rel1, rel2 , ens) ≡ ∀o : o ∈ ens ⇒ (
¬(Empty?(Filter(rel1, Sel1(rel1i = o)))) ⇒ Empty?(Filter(rel2, Sel2(rel2i = o))) ∧
¬(Empty?(Filter(rel2, Sel2(rel2i = o)))) ⇒ Empty?(Filter(rel1, Sel1(rel1i = o)))))
Cens2(rel1) = ens ∧ Cens1(rel2) = ens ⇒ (Exclusion(rel1, rel2 , ens) ≡ ∀o : o ∈ ens ⇒ (
¬(Empty?(Filter(rel1, Sel2(rel1i = o)))) ⇒ Empty?(Filter(rel2, Sel1(rel2i = o))) ∧
¬(Empty?(Filter(rel2, Sel1(rel2i = o)))) ⇒ Empty?(Filter(rel1, Sel2(rel1i = o)))))
Cens2(rel1) = ens ∧ Cens2(rel2) = ens ⇒ (Exclusion(rel1, rel2 , ens) ≡ ∀o : o ∈ ens ⇒ (
¬(Empty?(Filter(rel1, Sel2(rel1i = o)))) ⇒ Empty?(Filter(rel2, Sel2(rel2i = o))) ∧
¬(Empty?(Filter(rel2, Sel2(rel2i = o)))) ⇒ Empty?(Filter(rel1, Sel2(rel1i = o)))))
hints
o : OBJET
rel1, rel2 : RELATOB
ens : SET[<u>OBJET</u>]

OPERATION Simultanéité

arity $\underline{RELATOB} \times \underline{RELATOB} \times SET[\underline{OBJET}] \rightarrow BOOLEAN$

pré $(Cens1(rel1) = ens \vee Cens2(rel1) = ens) \wedge (Cens1(rel2) = ens \vee Cens2(rel2) = ens)$

post

$Cens1(rel1) = ens \wedge Cens1(rel2) = ens \Rightarrow (\text{Simultanéité}(rel1, rel2, ens) \equiv \forall o : o \in ens \Rightarrow ($
$\neg(Empty?(Filter(rel1, Sel1(rel1i = o)))) \Leftrightarrow \neg(Empty?(Filter(rel2, Sel1(rel2i = o))))))$

$Cens1(rel1) = ens \wedge Cens2(rel2) = ens \Rightarrow (\text{Simultanéité}(rel1, rel2, ens) \equiv \forall o : o \in ens \Rightarrow ($
$\neg(Empty?(Filter(rel1, Sel1(rel1i = o)))) \Leftrightarrow \neg(Empty?(Filter(rel2, Sel2(rel2i = o))))))$

$Cens2(rel1) = ens \wedge Cens1(rel2) = ens \Rightarrow (\text{Simultanéité}(rel1, rel2, ens) \equiv \forall o : o \in ens \Rightarrow ($
$\neg(Empty?(Filter(rel1, Sel2(rel1i = o)))) \Leftrightarrow \neg(Empty?(Filter(rel2, Sel1(rel2i = o))))))$

$Cens2(rel1) = ens \wedge Cens2(rel2) = ens \Rightarrow (\text{Simultanéité}(rel1, rel2, ens) \equiv \forall o : o \in ens \Rightarrow ($
$\neg(Empty?(Filter(rel1, Sel2(rel1i = o)))) \Leftrightarrow \neg(Empty?(Filter(rel2, Sel2(rel2i = o))))))$

hints

o : OBJET
rel1, rel2 : RELATOB
ens : SET[<u>OBJET</u>]

OPERATION Totalité

arity $\underline{RELATOB} \times \underline{RELATOB} \times SET[\underline{OBJET}] \rightarrow BOOLEAN$

pré $(Cens1(rel1) = ens \vee Cens2(rel1) = ens) \wedge (Cens1(rel2) = ens \vee Cens2(rel2) = ens)$

$Cens1(rel1) = ens \wedge Cens1(rel2) = ens \Rightarrow (\text{Totalité}(rel1, rel2, ens) \equiv \forall o : o \in ens \Rightarrow ($
$\neg(Empty?(Filter(rel1, Sel1(rel1i = o)))) \vee \neg(Empty?(Filter(rel2, Sel1(rel2i = o))))))$

$Cens2(rel1) = ens \wedge Cens1(rel2) = ens \Rightarrow (\text{Totalité}(rel1, rel2, ens) \equiv \forall o : o \in ens \Rightarrow ($
$\neg(Empty?(Filter(rel1, Sel2(rel1i = o)))) \vee \neg(Empty?(Filter(rel2, Sel1(rel2i = o))))))$

$Cens1(rel1) = ens \wedge Cens2(rel2) = ens \Rightarrow (\text{Totalité}(rel1, rel2, ens) \equiv \forall o : o \in ens \Rightarrow ($
$\neg(Empty?(Filter(rel1, Sel1(rel1i = o)))) \vee \neg(Empty?(Filter(rel2, Sel2(rel2i = o))))))$

$Cens2(rel1) = ens \wedge Cens2(rel2) = ens \Rightarrow (\text{Totalité}(rel1, rel2, ens) \equiv \forall o : o \in ens \Rightarrow ($
$\neg(Empty?(Filter(rel1, Sel2(rel1i = o)))) \vee \neg(Empty?(Filter(rel2, Sel2(rel2i = o))))))$

hints

o : OBJET
ens : SET[<u>OBJET</u>]
rel1, rel2 : RELATOB

OPERATION Inclusion

arity $\underline{RELATOB} \times \underline{RELATOB} \times \underline{OBJET} \rightarrow BOOLEAN$

pré $(Cens1(rel1) = ens \vee Cens2(rel1) = ens) \wedge (Cens1(rel2) = ens \vee Cens2(rel2) = ens)$

post

$Cens1(rel1) = ens \wedge Cens1(rel2) = ens \Rightarrow (\text{Totalité}(rel1, rel2, ens) \equiv \forall o : o \in ens \Rightarrow ($
$\neg(Empty?(Filter(rel1, Sel1(rel1i = o)))) \Rightarrow \neg(Empty?(Filter(rel2, Sel1(rel2i = o))))))$

$Cens1(rel1) = ens \wedge Cens2(rel2) = ens \Rightarrow (\text{Totalité}(rel1, rel2, ens) \equiv \forall o : o \in ens \Rightarrow ($
$\neg(Empty?(Filter(rel1, Sel1(rel1i = o)))) \Rightarrow \neg(Empty?(Filter(rel2, Sel2(rel2i = o))))))$

$Cens2(rel1) = ens \wedge Cens1(rel2) = ens \Rightarrow (\text{Totalité}(rel1, rel2, ens) \equiv \forall o : o \in ens \Rightarrow ($
$\neg(Empty?(Filter(rel1, Sel2(rel1i = o)))) \Rightarrow \neg(Empty?(Filter(rel2, Sel1(rel2i = o))))))$

$Cens2(rel1) = ens \wedge Cens2(rel2) = ens \Rightarrow (\text{Totalité}(rel1, rel2, ens) \equiv \forall o : o \in ens \Rightarrow ($
$\neg(Empty?(Filter(rel1, Sel2(rel1i = o)))) \Rightarrow \neg(Empty?(Filter(rel2, Sel2(rel2i = o))))))$

hints

o : OBJET
ens : SET[<u>OBJET</u>]
rel1, rel2 : RELATOB

END CONTRAINTE

Traduisons un schéma inspiré d'un exemple donné dans la présentation du formalisme (certaines notations ont été changées vu que nous ne les avons pas étudiées et le schéma a été simplifié pour que la traduction ne soit pas trop longue) :

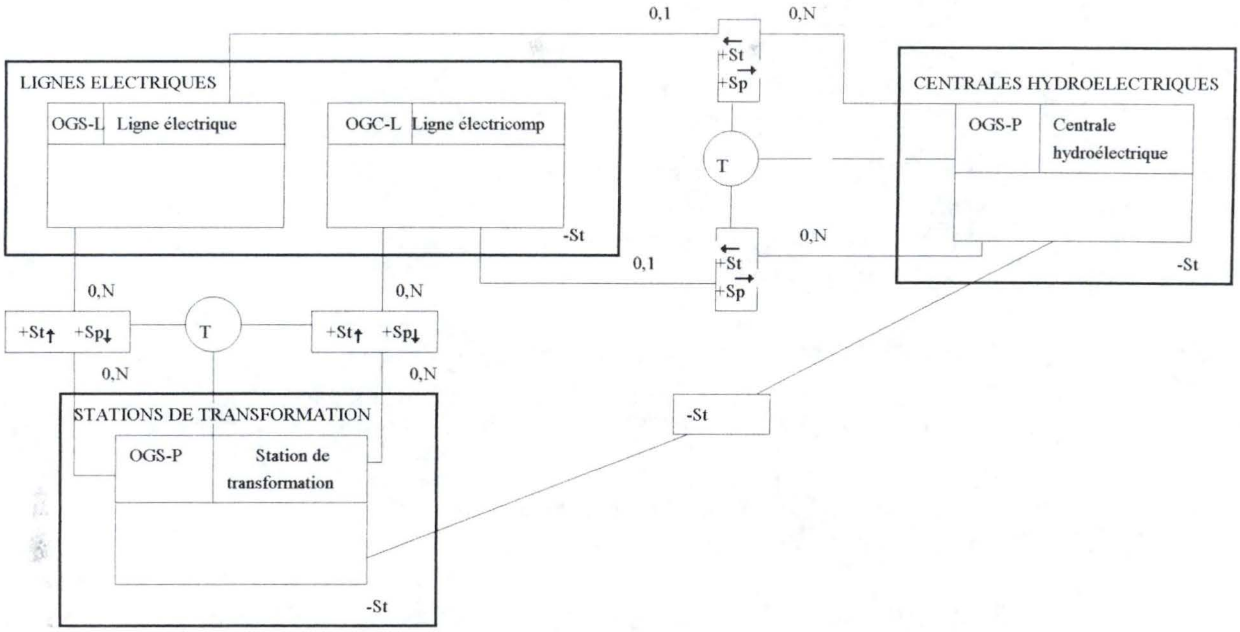


Figure : 3.1

Traduction :

SCHEMA = CP[L : LIGNESELECTRIQUES, C : CENTRALESHYDROELECTRIQUES, S : STATIONSDETRANSFORMATION, R1 : REL_TOPO_0-1_0-N[LIGNEELECTRIQUE, CENTRALEHYDROELECTRIQUE], R2 : REL_TOPO_0-1_0-N[LIGNEELECTRICOMP, CENTRALEHYDROELECTRIQUE], R3 : REL_TOPO_0-N_0-N[LIGNEELECTRIQUE, STATIONDETRANSFORMATION], R4 : REL_TOPO_0-N_0-N[LIGNEELECTRICOMP, STATIONDETRANSFORMATION], R5 : REL_TOPO-S[STATIONDETRANSFORMATION, CENTRALEHYDROELECTRIQUE]]

asserts : ∀s : Ens1(L(s)) = Cens1(R1(s)) ∧ Ens1(L(s)) = Cens1(R3(s)) ∧ Ens2(L(s)) = Cens1(R2(s)) ∧ Ens2(L(s)) = Cens1(R4(s)) ∧ Ens(C(s)) = Cens2(R1(s)) ∧ Ens(C(s)) = Cens2(R5(s)) ∧ Ens(S(s)) = Cens1(R5(s)) ∧ Ens(S(s)) = Cens2(R1(s)) ∧ Ens(S(s)) = Cens2(R3(s)) ∧ Ens(C(s)) = Cens2(R2(s)) ∧ Ens(S(s)) = Cens1(R5(s)) ∧ Totalité(R1(s), R2(s), Ens(C(s))) ∧ Totalité(R3(s), R4(s), Ens(S(s)))
Hints : s : SCHEMA

REL_TOPO_0-1_0-N[LIGNEELECTRIQUE, CENTRALEHYDROELECTRIQUE]
is REL_TOPO[LIGNEELECTRIQUE, CENTRALEHYDROELECTRIQUE]
asserts : ∀r : Card1(r) = '0,1' ∧ Card2(r) = '0,N' ∧ Vrel1(r) = 'rien' ∧ Vrel2(r) ∧ Srel1(r) = '←St' ∧ Srel2(r) = 'Sp→'
hints : r : REL_TOPO_0-1_0-N[LIGNEELECTRIQUE,CENTRALEHYDROELECTRIQUE]

REL_TOPO_0-1_0-N[LIGNEELECTRICOMP, CENTRALEHYDROELECTRIQUE]
is REL_TOPO[LIGNEELECTRICOMP, CENTRALEHYDROELECTRIQUE]
asserts : ∀r : Card1(r) = '0,1' ∧ Card2(r) = '0,N' ∧ Vrel1(r) = 'rien' ∧ Vrel2(r) ∧ Srel1(r) = '← St' ∧ Srel2(r) = 'Sp→'
hints : r : REL_TOPO_0-1_0-N[LIGNEELECTRICOMP, CENTRALEHYDROELECTRIQUE]

REL_TOPO_0-N_0-N[LIGNEELECTRIQUE, STATIONDETRANSFORMATION]
is REL_TOPO[LIGNEELECTRIQUE, STATIONDETRANSFORMATION]
asserts : ∀r : Card1(r) = '0,N' ∧ Card2(r) = '0,N' ∧ Vrel1(r) = 'rien' ∧ Vrel2(r) ∧ Srel1(r) = '← St' ∧ Srel2(r) = 'Sp→'
hints : r : REL_TOPO_0-N_0-N[LIGNEELECTRIQUE, STATIONDETRANSFORMATION]

REL_TOPO_0-N_0-N[LIGNEELECTRICOMP, STATIONDETRANSFORMATION]
is REL_TOPO[LIGNEELECTRICOMP, STATIONDETRANSFORMATION]
asserts : ∀r : Card1(r) = '0,N' ∧ Card2(r) = '0,N' ∧ Vrel1(r) = 'rien' ∧ Vrel2(r) ∧ Srel1(r) = '← St' ∧ Srel2(r) = 'Sp→'
hints : r : REL_TOPO_0-N_0-N[LIGNEELECTRICOMP, STATIONDETRANSFORMATION]

REL_TOPO-S[STATIONDETRANSFORMATION, CENTRALEHYDROELECTRIQUE]]
is REL_TOPO[STATIONDETRANSFORMATION, CENTRALEHYDROELECTRIQUE]]
asserts : ∀r : Card1(r) = 'rien' ∧ Card2(r) = 'rien' ∧ Vrel1(r) = 'rien' ∧ Vrel2(r) ∧ Srel1(r) = '← St' ∧ Srel2(r) = 'St→'
hints :
r : REL_TOPO-S[STATIONDETRANSFORMATION, CENTRALEHYDROELECTRIQUE]]

LIGNESELECTRIQUES is CLASSE[LIGNEELECTRIQUE, LIGNEELECTRICOMP]
asserts : ∀ cs : Vobli(cs) = 'rien' ∧ Vpermis(cs) = 'rien' ∧ Sobli(cs) = '-St' ∧ Spermis = '+Sn,p'
hints : cs : LIGNESELECTRIQUES
LIGNEELECTRIQUE is OGS_L
LIGNEELECTRICOMP is OGC_L

CENTRALESHYDROELECTRIQUES is CLASSE[CENTRALEHYDROELECTRIQUE]
asserts : ∀ cs : Vobli(cs) = 'rien' ∧ Vpermis(cs) = 'rien' ∧ Sobli(cs) = '-St' ∧ Spermis = 'rien'
hints : cs : CENTRALESHYDROELECTRIQUES
CENTRALEHYDROELECTRIQUE is OGS_P

STATIONSDETRANSFOMATION is CLASSE[STATIONDETRANSFORMATION]
asserts : ∀ cs : Vobli(cs) = 'rien' ∧ Vpermis(cs) = 'rien' ∧ Sobli(cs) = '-St' ∧ Spermis = 'rien'
hints : cs : STATIONSDETRANSFORMATION
STATIONDETRANSFOMATION is OGS_P

Le travail d'élaboration d'une sémantique formelle pour CONGOO que nous venons d'effectuer n'est pas terminé. En effet nous n'avons pas donné une sémantique formelle à tous les concepts de CONGOO. De plus il faudrait aussi pour achever ce travail avoir de nombreuses interactions avec l'auteur du modèle pour lever certaines ambiguités et valider les définitions que nous avons données.

Idéalement ce travail devrait se terminer par l'élaboration des règles de traduction d'un schéma CONGOO au langage GLIDER mais pour arriver à ce résultat il faudrait que le modèle CONGOO possède une syntaxe et une sémantique complètes et précises ce qui n'est pas le cas actuellement.

3.4 CONCLUSION

Bien que ce travail d'élaboration d'une sémantique formelle ne soit pas terminé, il montre l'apport et l'avantage de faire une spécification formelle du modèle CONGOO. En effet notre spécification formelle a permis de mettre en évidence quelques ambiguités ainsi que certaines incomplétudes dans les règles concernant les concepts. De plus elle permet maintenant d'avoir une interprétation univoque des concepts que nous avons étudiés. Une fois cette spécification formelle terminée elle permettra aussi de réaliser des outils "CASE" performants. Enfin les définitions formelles des relations topologiques et des éléments géométriques sur lesquels elle est appliquée vont nous permettre de déduire formellement les propriétés de ces relations et de raisonner. Ceci nous donnera la possibilité de vérifier et maintenir la cohérence entre les relations topologiques d'un schéma mais aussi de gérer la redondance de l'information topologique. Pour conclure, soulignons que l'utilisation d'un langage formel a un gros désavantage, c'est qu'il n'est pas compréhensible par tout le monde ou du moins qu'il demande une certain effort pour le comprendre.

Chapitre 4
Les relations topologiques

4.1. Introduction

Dans le chapitre deux, nous avons vu qu'une des grandes originalités du modèle CONGOO est d'avoir intégré des relations spécifiques pour exprimer les contraintes topologiques entre les objets géographiques. A présent, nous allons étudier plus en détail ces relations topologiques. Tout d'abord nous tentons de définir ce qu'on entend par les relations topologiques en géomatique et de préciser l'utilité de stocker l'information topologique dans une base de données d'un système d'information géographique. Ensuite, vu que la diversité, la sémantique, la terminologie et la complétude de ces relations varient dramatiquement dans la littérature scientifique, nous avons décidé de reprendre brièvement les travaux qui existent sur ce sujet afin de pouvoir juger de la pertinence du choix des relations topologiques utilisées dans le modèle CONGOO. Après, nous présentons une définition formelle de ces relations basée sur les notions de l'algèbre topologique. Cette définition formelle nous a permis de comparer facilement les relations topologiques du modèle CONGOO à celle définie dans les autres travaux et de déduire leurs propriétés. Enfin, dans la dernière partie de ce chapitre, nous étudions formellement la composition des relations topologiques. Cette étude nous semble fort importante car elle est à notre avis le fondement de la recherche pour contrôler la redondance ou vérifier la cohérence d'un schéma CONGOO au niveau topologique.

4.2. Définition

Avant de définir ce qu'on entend par relation topologique en géomatique, nous allons tenter de donner la signification du terme topologie. Comme le signalent divers ouvrages [Barr 87], [topo 81] la topologie est une branche des mathématiques mais, contrairement aux autres branches (l'arithmétique, la géométrie,..), il est très difficile de donner une définition précise de ce terme en quelques mots. Nous nous contenterons donc de faire comprendre ce qu'étudie le vaste champ de la topologie.

En un sens, nous pouvons dire que la topologie est l'étude de la continuité : commençant avec la continuité de l'espace, elle se généralise et conduit alors par analogie à d'autres sortes de continuité. Dans l'espace, cette discipline s'intéresse aux propriétés d'une chose qui sont les plus permanentes, c'est-à-dire celles qui survivent à des distorsions et des étirements. Par exemple en topologie, une ligne droite n'a pas à rester droite mais elle conserve la qualité d'être continûment connexe le long d'elle-même. C'est cette connexité, cette continuité que la topologie privilégie et c'est pour cette raison que des distorsions sont autorisées, seulement si aucune ne déconnecte ce qui était connexe, ni ne connecte ce qui ne l'était pas.

Un concept central de la théorie de la topologie est l'équivalence topologique. "Deux figures x et y sont dites homéomorphes (topologiquement équivalentes) s'il existe une application bijective continue f de x sur y dont la réciproque f^{-1} soit aussi continue. L'application f est alors un homéomorphisme"[math 80, p.740]. En d'autres termes, deux figures x et y sont homéomorphes si x peut être comprimée ou déformée ou étirée en y sans que deux parties de x soient déchirées ou collées. Les propriétés des figures topologiques qui sont préservées sous homéomorphisme sont appelées invariants topologiques.

Les relations topologiques utilisées en géomatique sont un sous-ensemble spécifique des relations spatiales. Elles sont caractérisées par la propriété d'être préservée sous les transformations topologiques telles que rotation, translation, changement d'échelle de deux objets géographiques c'est-à-dire qu'il s'agit des relations spatiales qui sont préservées sous homéomorphisme. Ainsi par exemple les chemins sont adjacents aux prés, la maison au numéro 5 est voisine des maisons numéro 3 et 7, ou encore la ville de Bruxelles est incluse dans la région flamande, sont des relations topologiques entre deux objets géographiques. Remarquons donc qu'une relation topologique entre deux objets se réfère à leurs positions relatives dans l'espace, en abstraction de distance, forme et grandeur.

Nous pouvons définir l'équivalence de deux relations topologiques. La relation topologique entre l'objet X1 et l'objet Y1 dans l'espace topologique A est équivalente à celle entre l'objet X2 et l'objet Y2 dans l'espace topologique B s'il y a un homéomorphisme f de A à B tel que f(X1) = X2 et f(Y1) = (Y2).

4.3. Pourquoi stocker l'information topologique dans une base de données ?

Les relations topologiques entre les objets géographiques peuvent bien sûr être déterminées par leurs positions absolues (en terme de coordonnées) mais en général elles sont stockées explicitement dans la base de données. Trois raisons principales peuvent être relevées [Hadzilacos et Tryfona 92].

Tout d'abord, il peut être tout simplement parfois nécessaire de manipuler topologiquement des objets avant que les distances, formes et grandeurs soient établies. Les représentations quantitatives ont toujours besoin d'une description complète de l'objet géométrique, et il y a de sérieux problèmes quand l'information géométrique est imprécise.

De plus, la somme des traitements requis pour extraire les relations topologiques entre les objets de leurs positions absolues est très lourde. Par contre la somme d'informations dont on a besoin pour décrire la topologie est souvent peu importante.

Enfin, dans les bases de données traditionnelles, on exprime des contraintes d'intégrités pour contrôler si l'état ou la transition d'un état à un autre de la base de données est correcte. Dans les systèmes d'information géographique, il faut en plus tenir compte de la topologie pour déterminer si l'état des bases de données est correct ou pas. En effet par exemple deux pays voisins doivent avoir une limite en commun ; une maison ne peut être bâtie dans une zone forestière ; un lac et une mer ne peuvent pas avoir d'intersection. Soulignons qu'il serait totalement impraticable et incommode d'exprimer les contraintes d'intégrités topologiques en termes de position absolue.

C'est donc pour des raisons d'efficacité et d'intégrité que les relations topologiques sont habituellement stockées dans les bases de données des systèmes d'information géographique malgré leur redondance avec les données absolues.

4.4. Critiques des travaux sur les relations topologiques

Dans la littérature informatique et géographique, une multitude de termes variés peuvent être trouvés pour nommer les relations spatiales [Chang, Jeugert et Li 89], [Clementini, Di Felice et van Oosterom 93], [Laurini et Milleret-Raffort 93] et notamment dans les langages d'interrogation spatiale , [de Hoop, van Oosterom et Molenaar 93], [Smith 87]. La plupart du temps ces termes n'ont pas de sémantique précise. En effet ils sont souvent définis en langage naturel, ce qui laisse souvent des ambiguïtés car leur définition est exprimée avec des termes n'ayant pas de définition claire et précise.

Il n'existe que très peu d'approches formelles sur les relations topologiques. Pourtant comme le rappelle Egenhofer [Egenhofer 89], l'approche formelle est très bénéfique pour différentes raisons : " 1.The formalism serves as a tool to identify and derive relationships. Redundant and contradicting relations can be avoided such that a minimal set of fundamental relationships can be defined. 2.The formal methods can be applied to determine the relation between any two spatial objects. Algorithms to determine relationships can be specified exactly, and mathematically sound models will help to define formally the relationships. 3.The formalism is expected to help prove the completeness of the set of relationships. 4.The fundamental relations can be used to combine more complex relations ."

Reprenons et critiquons à présent les principaux travaux basés sur une approche formelle. Comme nous l'avons déjà dit dans l'introduction, ce bref exposé sur les différents travaux nous permettra de critiquer et de comparer facilement la solution de CONGOO avec ceux-ci et d'améliorer, si nécessaire, cette solution.

4.4.1. Les recherches basées sur les travaux d'Allen

Plusieurs chercheurs se sont inspirés fortement ou faiblement du très bon article d'Allen "Maintaining Knowledge about Temporal Intervals " [Allen 83] introduisant une logique temporelle basée sur les intervalles pour l'étendre aux dimensions spatiales. Une manière de procéder a été de donner le même type de distinction que pour le cas temporal aux deux axes du système de coordonnées cartésiennes. L'objet géographique est alors projeté sur les deux axes et un tuple cartésien est utilisé pour donner la position relative de l'objet. A propos de ce travail nous rejoignons l'avis d'Egenhofer "People don't walk around decomposing the world into axes and then determining a qualitative relation for an interval on each of them " [Egenhofer et Herring 95]. En effet l'inconvénient majeur est que cette approche est difficile à lire et peu intuitive .

4.4.2. Les travaux basés sur la théorie des ensembles

Dans le contexte d'une algèbre géo-relationnelle, Güting [Güting 88] donne une définition formelle aux relations spatiales. Dans cette approche les objets géographiques sont représentés sous forme d'ensembles de points et les relations topologiques sont décrites en terme des opérateurs conventionnels des ensembles par comparaison des ensembles de points de deux objets géographiques. Ainsi par exemple les relations "equal", "not equal", "inside", "outside" et "intersects" ont été définies :

x equal y := Points (x) = Points (y)
x not équal y := Points(x) $\neq$ Points (y)
x inside y := Points (x) $\subseteq$ Points (y)
x outside y := Points (x) $\cap$ Points (y) = $\varnothing$
x intersects y := Points (x) $\cap$ Points (y) $\neq$ $\varnothing$

Cette solution bien que fort attrayante au premier abord notamment par sa simplicité, a de nombreux inconvénients. Tout d'abord, les définitions de ces ensembles de relations ne sont pas orthogonales : ainsi par exemple la définition "Equal" et "Inside" sont couvertes toutes les deux par la définition "Intersects". Enfin, cet ensemble de relations n'est pas complet et la théorie pure d'ensembles de points ne permet pas de l'être. En effet la relation très importante de voisinage entre deux objets ne peut être définie (appelée aussi relation adjacente, meet..) car la caractéristique cruciale de cette relation est que les objets géographiques ont une partie de leurs frontières en commun et pas leurs intérieurs.

Pour remédier à ce dernier inconvénient Pullar [Pullar 88], a ajouté les concepts de frontière et d'intérieur à l'approche des ensembles de points, ce qui a permis de distinguer "overlap" et "neighbor" .Une autre approche [Wagner 88] plus systématique sur l'intersection de la frontière et de l'intérieur a permis d'identifier quatre relations :

x neighborhood y := Boundary (x) $\cap$ Boundary (y) $\neq \varnothing \wedge$ Intérieur (x) $\cap$ Intérieur (y) $= \varnothing$..
x intersection y := Boundary (x) $\cap$ Boundary (y) $\neq \varnothing \wedge$ Intérieur (x) $\cap$ Intérieur (y) $\neq \varnothing$..
x inclusion y := Boundary (x) $\cap$ Boundary (y) $= \varnothing \wedge$ Intérieur (x) $\cap$ Intérieur (y) $\neq \varnothing$..
x séparation y := Boundary (x) $\cap$ Boundary (y) $= \varnothing \wedge$ Intérieur (x) $\cap$ Intérieur (y) $= \varnothing$..

Un problème de cette formalisation est qu'elle n'autorise pas la distinction entre l'intersection et l'égalité de deux objets .

4.4.3. Les travaux basés sur l'algèbre topologique

Une des plus populaires des approches formelles pour décrire les relations topologiques qui peut être trouvée dans la littérature est celle d'Egenhofer [Egenhofer 89], [Egenhofer et Franzosa 91], [Egenhofer et Herring 95]. Sa théorie est basée sur les notions topologiques d'intérieur et de frontière. Les objets géographiques sont représentés sous forme d'ensembles de points et les relations topologiques sont définies grâce aux intersections de frontière (∂A) et intérieur (A°) de deux ensembles. Ainsi en considérant vide et non vide comme les seules valeurs possibles de l'intersection, un total de 16 relations (Tableau 1) mutuellement exclusives peuvent être réalisées dans R^2. Cet ensemble est réduit à 8 relations (Figure 1) si les régions spatiales considérées satisfont à (1)A est connecté, (2)$A = \tilde{A}^\circ$, (3)$\partial A \neq \varnothing$.

	$\partial \cap \partial$	$^\circ \cap ^\circ$	$\partial \cap ^\circ$	$^\circ \cap \partial$
r0	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$
r1	$\neg\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$
r2	$\varnothing$	$\neg\varnothing$	$\varnothing$	$\varnothing$
r3	$\neg\varnothing$	$\neg\varnothing$	$\varnothing$	$\varnothing$
r4	$\varnothing$	$\varnothing$	$\neg\varnothing$	$\varnothing$
r5	$\neg\varnothing$	$\varnothing$	$\neg\varnothing$	$\varnothing$
r6	$\varnothing$	$\neg\varnothing$	$\neg\varnothing$	$\varnothing$
r7	$\neg\varnothing$	$\neg\varnothing$	$\neg\varnothing$	$\varnothing$
r8	$\varnothing$	$\varnothing$	$\varnothing$	$\neg\varnothing$
r9	$\neg\varnothing$	$\varnothing$	$\varnothing$	$\neg\varnothing$
r10	$\varnothing$	$\neg\varnothing$	$\varnothing$	$\neg\varnothing$
r11	$\neg\varnothing$	$\neg\varnothing$	$\varnothing$	$\neg\varnothing$
r12	$\varnothing$	$\varnothing$	$\neg\varnothing$	$\neg\varnothing$
r13	$\neg\varnothing$	$\varnothing$	$\neg\varnothing$	$\neg\varnothing$
r14	$\varnothing$	$\neg\varnothing$	$\neg\varnothing$	$\neg\varnothing$
r15	$\neg\varnothing$	$\neg\varnothing$	$\neg\varnothing$	$\neg\varnothing$

Tableau1 : Les 16 relations topologiques binaires basées sur les valeurs de vide et non vide des intersections des frontières et des intérieurs [Egenhofer et Franzosa 91]

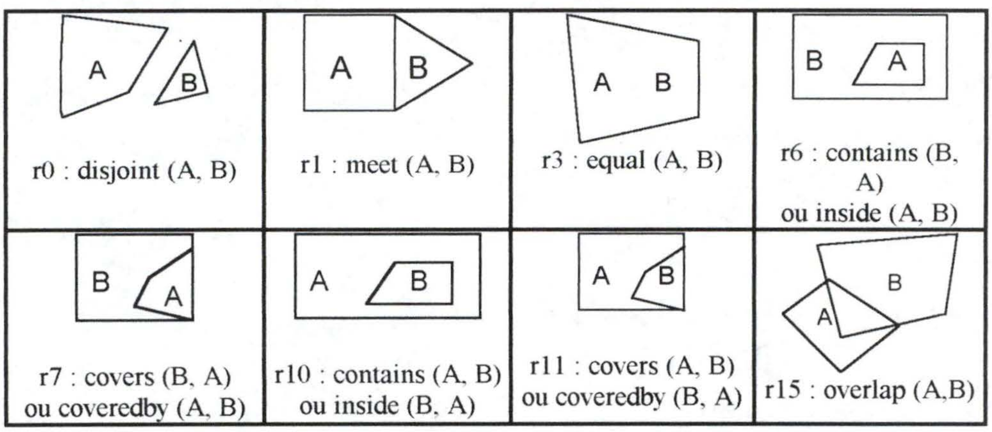


Figure 1 : Représentation et terminologie des huit relations entre deux régions spatiales

Remarquons que les relations "disjoint", "meet", "equal", "overlap" sont des relations symétriques et deux paires de relations ont un inverse "covers", "coveredBy" et "inside", "contains" .

Ce modèle est à la base de très nombreuses études théoriques très avancées telles que la dérivation formelle de la table de composition pour cette ensemble de relations [Egenhofer 91], une comparaison de la similarité des relations topologiques dans un modèle vecteur et raster [Egenhofer et Sharma 93], une étude des relations topologiques entre des régions possédant des "trous" [Egenhofer, Clementini et Di Felice 94], le traité sur l'équivalence des relations topologiques [Egenhofer et Franzosa 95], le raisonnement autour du changement graduel d'une relation topologique causé par une translation, une rotation, ou un changement de taille d'un objet [Egenhofer et Al-Taha 92].

Par la suite l'auteur apporta une amélioration au modèle : il utilisa l'intersection des frontières et des intérieurs mais aussi des compléments des objets pour décrire leurs relations topologiques [Egenhofer 91]. Cette extension est notamment nécessaire pour juger si un objet est ou n'est pas complètement inclus dans un autre si les dimensions de ceux-ci ne sont pas les mêmes. Ainsi ce nouveau modèle a permis de développer un ensemble de relations plus complet entre des objets représentés par une ligne et une région [Egenhofer et Mark 95].

Ces très nombreux travaux jouissent d'une popularité considérable dans la recherche et sont très utilisés. Ainsi par exemple des concepteurs de systèmes d'information géographique et de systèmes de gestion de base de données l'ont utilisé [Herring 91]. Ils ont aussi été exploités pour concevoir un langage de query topologique [de Hoop et Van Oosteron 92]. D'autres chercheurs les utilisèrent comme base pour des recherches [Molenaar, Kufanigi et Bouloucos 94]. Certains [Clementini, Di Felice et van Oosterom 93], [Hadzilacos et Tryfona 92], essayèrent de généraliser le modèle de relation topologique entre point, ligne et région dans R^2 ou encore d'appliquer le modèle à des objets de trois ou quatre dimensions.

Ce modèle peut être critiqué pour différentes raisons :

1. Comme le signale Clementini [Clementini, Di Felice et van Oosterom 93], vu que le modèle considère seulement le vide ou le non vide comme valeur d'intersection, il ne permet pas de distinguer certaines situations pour lesquelles les personnes ont d'habitude des images mentales distinctes. Clementini propose de considérer la dimension de l'intersection pour résoudre ce problème. Ce choix devra en tous cas être optimum entre la complétude et la complexité d'utilisation qu'elle crée .

2. Un autre inconvénient majeur est le très grand nombre de relations différentes portant chacune un nom pour décrire la topologie. En effet nous avons vu qu'il y avait déjà huit relations différentes entre deux régions et il y en a dix-neuf entre des objets lignes et régions. Vu ce grand nombre de relations, il est difficile pour un utilisateur de se rappeler tous ces noms et leurs sémantiques, il engendre donc souvent une mauvaise utilisation due à une confusion .

3. Le modèle est basé sur des concepts mathématiques relativement compliqués tels que régions ouvertes, fermées, frontières pas toujours bien connus par l'utilisateur.

4.4.4. Les recherches basées sur les travaux de Clarke

Depuis plusieurs années, pour représenter et raisonner autour de l'information spatiale qualitative les chercheurs du département d'intelligence artificielle de l'université de Leeds effectuent des recherches par une approche axiomatique. Le résultat de ces recherches est paru dans de très nombreux articles [Randell, Cui et Cohn 92], [Cui, Cohn et Randell 93], [Cohn et Gotts 94]. Cette approche est basée sur les deux très bons articles de Clarke "A Calculus of Individuals Based on 'Connection' "[Clarke 81] et "Individuals and Point "[Clarke 85].

La base du formalisme est assurée par une relation dyadique primitive : C(x, y). Elle est lue comme "x connecté avec y" ou encore "x et y partage un point en commun". Les individus peuvent être interprétés comme des régions spatiales ou des régions temporelles. Bien sûr nous nous intéressons ici aux individus considérés comme des régions spatiales.

La relation C(x, y) est réflexive et symétrique. Deux axiomes sont introduits.

$\forall x\ C(x, x)$
$\forall xy\ [C(x,y) \Rightarrow C(x, y)]$

En utilisant la relation primitive, un ensemble de relations dyadiques peut être défini pour décrire les différentes situations topologiques sur deux régions spatiales.

' x is disconnected from y'	$DC(x, y) \equiv \neg\ C(x, y)$
'x is a part of y'	$P(x, y) \equiv \forall z[C(z, x) \Rightarrow C(z, y)]$
'x is a proper part of y'	$PP(x, y) \equiv P(x, y) \wedge \neg P(y, x)$
'x is identical with y'	$x = y \equiv P(x, y) \wedge P(y, x)$
'x overlaps y'	$O(x, y) \equiv \exists z\ [P(z, x) \wedge P(z, y)]$
'x partially overlaps y'	$PO(x, y) \equiv O(x, y) \wedge \neg P(x, y) \wedge \neg P(y, x)$
'x is discrete from y'	$DR(x, y) \equiv \neg O(x, y)$
'x is externally connected with y'	$EC(x, y) \equiv C(x, y) \wedge \neg O(x, y)$
'x is a tangential proper part of y'	$TPP(x, y) \equiv PP(x, y) \wedge \exists z\ [EC(z, x) \wedge EC(z, y)]$
'x is a nontangential proper part of y'	$NTPP(x, y) \equiv PP(x, y) \wedge \neg\exists z\ [EC(z, x) \wedge EC(z, y)]$
	$P^{-1}(x, y) \equiv P(y, x)$
	$PP^{-1}(x, y) \equiv PP(y, x)$
	$TPP^{-1}(x, y) \equiv TPP(y, x)$
	$NTPP^{-1}(x, y) \equiv NTPP(y, x)$

Soulignons que le sous-ensemble des relations mutuellement exclusives {DC, EC, PO, =, TPP, NTTP, TPP^{-1}, $NTPP^{-1}$} correspond exactement à l'ensemble des huit relations topologiques entre deux régions trouvées par Egenhofer (Figure 2). Néanmoins, remarquons qu'il y a deux différences majeures par rapport au formalisme d'Egenhofer. Tout d'abord, le concept de base est la région loin de l'abstraction de l'ensemble de points qui peut sembler fort arbitraire. Ensuite, bien que région fermée et région ouverte peuvent être définies et utilisées comme dans le formalisme de Egenhofer, ils ont choisi de ne pas le faire car ils pensent, et à notre avis à juste titre, que le raisonnement sur les objets physiques et sur le sens commun n'a pas besoin d'une distinction entre région ouverte et fermée .

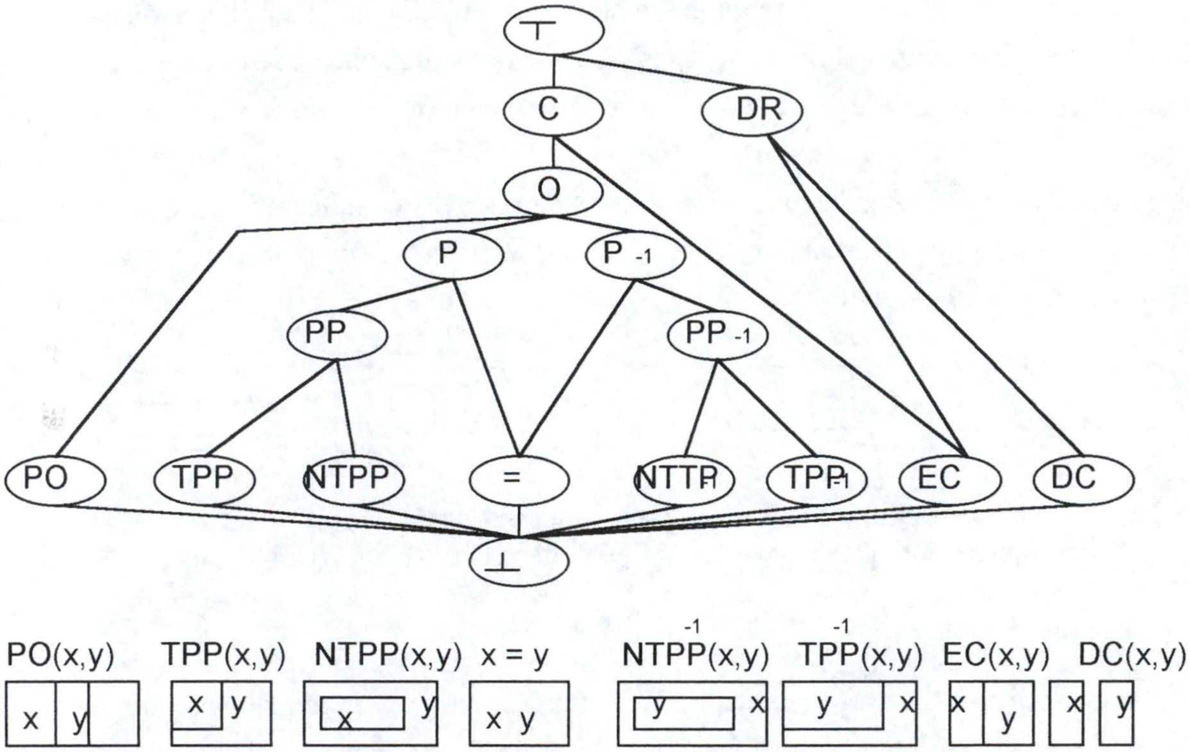


Figure 2 : hiérarchie des relations dyadiques définies en terme de la relation primitive C(x,y).

Nous avons relevé deux inconvénients importants à ce modèle. Premièrement, bien qu'il soit fondé sur une seule primitive il y a un grand nombre de définitions de relation à manipuler pour décrire la topologie entre les régions, ce qui est source de confusion. Deuxièmement, l'ensemble des relations de ce modèle sont seulement cohérentes entre des régions ayant la même dimension et au minimum deux. Ce modèle ne permet donc pas d'avoir des relations entre des objets de dimensions différentes et de donner une relation pour des objets représentés par un point ou une ligne, or cela est très important dans la modélisation de l'information géographique .

Signalons que d'autres chercheurs de différents laboratoires d'intelligence artificielle présentent des constructions logiques similaires fondées aussi sur les travaux de Clarke. Ainsi par exemple à l'université de Toulouse le travail de Vieu [Vieu 93].

De tous ces travaux relevons qu'aucun ne définit un ensemble complet de relations topologiques. Dans les modèles que nous avons vus, si deux relations sont différentes cela implique qu'elles ne sont pas des relations topologiques équivalentes (relations distinctes topologiquement) mais l'inverse n'est pas vrai. En effet si dans un modéle deux relations sont identiques elles ne sont pas nécessairement des relations topologiques équivalentes (Figure 3). Ceci suscite plusieurs questions pour lesquelles nous n'avons pas trouvé de réponses dans la littérature. Faut-il chercher à faire un ensemble "complet" de relations topologiques où deux relations identiques impliqueraient qu'elles soient topologiquements équivalentes? Nous pensons à ce propos que l'ensemble devra être optimun entre la complétude et la complexité d'utilisation qu'elle engendre et surtout que les situations pour lesquelles les personnes ont d'habitude des images mentales distinctes soient reprises dans cette ensemble. Quels sont les invariants topologiques prioritaires? Tout ceci montre qu'un travail important reste à faire et notamment au niveau cognitif.

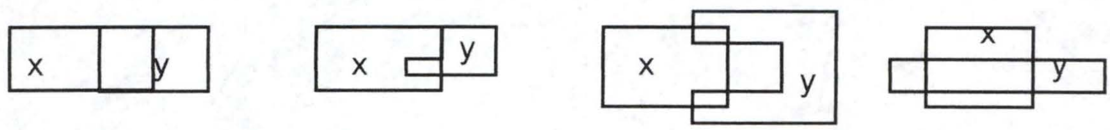


Figure 3 : Quatre relations distinctes topologiquement mais dans les modèles ce sont des relations identiques (Güting : x intersects y, Egenhofer : overlap(x,y), Cohn, Vieu.. : PO(x,y)).

Nous avons vu que chaque approche définissait un très grand nombre de relations différentes rendant leur utilisation difficile. De plus ces relations étaient principalement définies pour des régions à deux dimensions. Ceci souligne encore qu'il reste beaucoup de recherches à effectuer dans ce domaine et principalement dans la définition de relations entre objets complexes, de dimensions différentes et entre ensembles d'objets.

4.5. Formalisation des relations topologiques du modèle CONGOO basée sur les notions de l'algèbre topologique

Afin de faire une comparaison aisée avec les travaux vus ci-dessus, nous formalisons les relations topologiques du modèle CONGOO en nous basant sur les notions utilisées dans les travaux d'Egenhofer c'est-à-dire sur les notions d'intérieur, de frontière, de région ouverte, de région fermée.

Nous reprenons les notations et les définitions de Egenhofer [Egenhofer et Franzosa 91] d'intérieur ($x°$), frontière (∂x), fermeture (x). Nous formalisons tout d'abord les relations topologiques pour les objets représentés par un polygone.

Traduisons les relations entre intérieur, frontière et fermeture d'un polygone :

$\forall$ x : $x° \cup \partial x = x$

$\forall$ x : $x° \cap \partial x = \emptyset$

x : POLYGONE

Rappelons que dans le chapitre trois il nous a semblé important de faire la distinction entre l'objet et sa représentation. Pour ne pas surcharger l'écriture de la définition formelle nous travaillons au niveau de la représentation de l'objet dans la suite du travail. Remarquons que cela ne change pas grand chose puisque la superposition de deux objets est équivalente à la superposition de leurs représentations dans notre formalisation.

Définition des relations topologiques :

$\forall$ x,y : POLYGONE

Def1 : Superposition de x sur y ; $S(x,y) \equiv (x \cap y \neq \emptyset \wedge x \cap y \neq \partial x \cap \partial y)$

Def2 : Superposition totale de x sur y ; $St(x,y) \equiv x \cap y = x$

Def3 : Superposition partielle de x sur y $Sp(x,y) \equiv S(x,y) \wedge \neg St(x, y)$

Def4 : : Superposition nulle de x sur y ; $Sn(x,y) \equiv \neg S(x,y)$

Def5 : Voisinage de x sur y ; $V(x,y) \equiv \partial x \cap \partial y \neq \emptyset$

Def6 : Voisinage total de x sur y ; $Vt(x,y) \equiv \partial x \cap \partial y = \partial x$

Def7 : Voisinage partiel de x sur y ; $Vp(x,y) \equiv (\partial x \cap \partial y \neq \emptyset \wedge \neg(\partial x \cap \partial y = \partial x))$

$\equiv V(x,y) \wedge \neg Vt(x, y)$

Def8 : Voisinage nul de x sur y ; $Vn(x,y) \equiv \partial x \cap \partial y = \emptyset \equiv \neg V(x,y)$

La conjonction des deux relations de base (plus précisément des six relations) permet de retrouver les huit situations d'Egenhofer pour les régions (Tableau 4). Le modèle a donc l'avantage de réduire le nombre de concepts à manipuler et ainsi de diminuer la difficulté de maîtrise de ceux-ci.

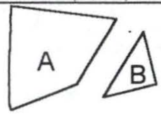
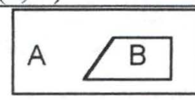
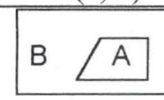
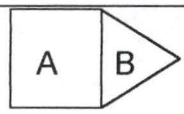
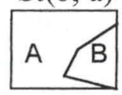
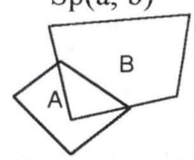
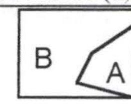
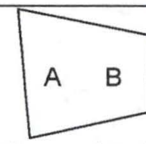
	Sn(a, b)	Sp (a, b)	St(a, b)
Vn(a, b)	 r0 : disjoint (a, b)	 r10 : contains (a, b) ou inside (b, a)	 r6 : contains (b, a) ou inside (a, b)
Vp(a, b)	 r1 : meet (a, b)	St(b, a)  r11 : covers (a, b) Sp(a, b)  r15 : overlap (a,b)	 r7 : covers (b, a) ou coveredby (a, b)
Vt(a, b)			 r3 : equal (a, b)

Tableau 4 : Les huit relations définies entre régions par Egenhofer exprimées par les relations de voisinages et de superpositions.

Maintenant si nous voulons "généraliser " les relations de voisinages et de superpositions aux objets représentés par un point ou une ligne nous devons compléter la formalisation. Pour pouvoir définir plus généralement les relations, nous introduisons un type représentation_simple qui est soit un type point, soit un type ligne, soit un type polygone et trois prédicats permettant de savoir si la variable de type représentation_simple est de type point ou de type ligne ou de type polygone.

$\forall$x : REPRESENTATION_SIMPLE
Point(x) $\equiv$ vrai si x est de type point
Ligne(x) $\equiv$ vrai si x est de type ligne
Polygone(x) $\equiv$ vrai si x est de type polygone

Rappelons-nous que dans ce modèle les représentations de type polygone ont une frontière (limite); par contre les représentations de type ligne et point n'en ont pas. Les formalismes trouvés dans la littérature [Egenhofer et Mark 95 ; Clementini, Di Felice, van Oosterom 93] donnent en général une frontière formée de deux points aux lignes sauf si elles sont fermées.

Si nous considérons que les lignes et les points ne possèdent pas de frontière et que les formules $\forall$ x : $x^\circ \cup \partial x = x$ et $\forall$ x : $x^\circ \cap \partial x = \varnothing$ sont vraies, nous pouvons dire que pour les lignes et les points $x^\circ = x$ et $\partial x = \varnothing$.

D1 : Superposition de x sur y : S(x, y)

$(\text{Polygone}(x) \wedge \text{Polygone}(y)) \Rightarrow S(x,y) \equiv (x \cap y \neq \varnothing \wedge x \cap y \neq \partial x \cap \partial y)$

$\neg(\text{Polygone}(x) \wedge \text{Polygone}(y)) \Rightarrow S(x,y) \equiv x^{\circ} \cap y^{\circ} \neq \varnothing$

D2 : Superposition totale de x sur y : St(x, y)

Précondition : $\neg((\text{Polygone}(x) \wedge ((\text{Point}(y) \vee \text{Ligne}(y))) \vee(\text{Ligne}(x) \wedge \text{Point}(y)))$

$(\text{Polygone}(x) \wedge \text{Polygone}(y)) \vee ((\text{Point}(x) \vee \text{Ligne}(x)) \wedge \text{Ligne}(y)) \Rightarrow St(x, y) \equiv x \cap y = x$

$(\text{Point}(x) \vee \text{Ligne}(y)) \wedge \text{Polygone}(y) \Rightarrow St(x, y) \equiv x \cap y = x \wedge S(x, y)$

D3 : Superposition partielle de x sur y : Sp(x, y)

Précondition : $\neg(\text{Point}(x))$

$((\text{Polygone}(x) \wedge ((\text{Point}(y) \vee \text{Ligne}(y))) \vee (\text{Ligne}(x) \wedge \text{Point}(y))) \Rightarrow Sp(x, y) \equiv S(x, y)$

$((\text{Ligne}(x) \vee \text{Polygone}) \wedge \text{Polygone}(y)) \Rightarrow Sp(x, y) \equiv S(x, y) \wedge \neg(St(x, y))$

D4 : superposition nulle de x sur y : $Sn(x, y) \equiv \neg\, S(x, y)$

D5 : Voisinage de x sur y :

précondition : $\text{Polygone}(x) \vee \text{Polygone}(y)$

$\text{Polygone}(x) \wedge (\text{Point}(y) \vee \text{Ligne}(y)) \Rightarrow V(x, y) \equiv \partial x \cap y \neq \varnothing$

$\text{Polygone}(y) \wedge (\text{Point}(x) \vee \text{Ligne}(x)) \Rightarrow V(x, y) \equiv \partial y \cap x \neq \varnothing$

$\text{Polygone}(x) \wedge \text{Polygone}(y) \Rightarrow V(x, y) \equiv \partial x \cap \partial y \neq \varnothing$

D6 : Voisinage total de x sur y :

précondition : $\text{Polygone}(x) \vee \text{Polygone}(y) \wedge \neg(\text{point}(y))$

$\text{Polygone}(x) \wedge \text{Ligne}(y) \Rightarrow Vt(x, y) \equiv \partial x \cap y = \partial x$

$\text{Polygone}(y) \wedge (\text{Point}(x) \vee \text{Ligne}(x)) \Rightarrow Vt(x, y) \equiv \partial y \cap x = x$

$\text{Polygone}(x) \wedge \text{Polygone}(y) \Rightarrow Vt(x, y) \equiv \partial x \cap \partial y = \partial x$

D7 : Voisinage partiel de x sur y :

précondition : $\text{Polygone}(x) \vee \text{Polygone}(y) \wedge \neg(\text{point}(x))$

$\text{Polygone}(x) \wedge \text{Point}(y) \Rightarrow Vp(x, y) \equiv V(x, y)$

$(\text{Polygone}(x) \vee \text{Ligne}(x)) \wedge (\text{Polygone}(y) \vee \text{Ligne}(y)) \Rightarrow Vp(x, y) \equiv V(x, y) \wedge \neg(Vt(x, y))$

D8 : Voisinage nul de x sur y :

précondition : $\text{Polygone}(x) \vee \text{Polygone}(y)$

$Vn(x, y) \equiv \neg V(x,y)$

Nous pouvons voir grâce à ces définitions que les relations de S, de V, de Vn et de Sn sont symétriques et que les relations de S et de St sont réflexives pour tous les types de représentations. La relation de St entre deux points est aussi symétrique. Pour les polygones, les relations de V et de Vt sont réflexives. Nous pouvons déduire aussi des formules et des propriétés des éléments géométriques que :

$St(x, y) \Rightarrow Sp(y, x) \vee St(y, x)$

$Sp(x, y) \Rightarrow Sp(y, x) \vee St(y, x)$

$\forall x, y : Polygone(x) \wedge Polygone(y) \wedge Vt(x, y) \Rightarrow St(x, y)$

$\forall$ x, y : $(Point(x) \vee Ligne(x)) \wedge Polygone(y) \wedge Vt(x, y) \Rightarrow Sn(x, y)$

$\forall$ x, y : $Point(x) \wedge Polygone(y) \wedge St(x, y) \Rightarrow Vn(x, y)$

$\forall$ x, y : $Ligne(x) \wedge Polygone(y) \wedge Sp(x, y) \Rightarrow Vp(x, y)$

$\forall$ x, y : $Ligne(x) \wedge Polygone(y) \wedge St(x, y) \Rightarrow \neg(Vt(x, y))$

Remarquons pour terminer ce petit exposé sur les propriétés des relations topologiques que les relations de Sn, de Sp et de St sont mutuellements exclusives ainsi que les relations de Vn, Vp et de Vt.

Le choix des relations topologiques du modèle CONGOO pour décrire les différentes situations topologiques entre objets est très intéressant car les définitions de ces relations topologiques sont relativements intuitives. De plus ces relations sont peu nombreuses et permettent de détailler un très grand nombre de situations topologiques différentes. Ces relations topologiques paraissent d'une utilisation plus facile pour caractériser une situation topologique que celles des autres travaux vu ci-dessus.

4.6. Etude de la composition des relations topologiques

Dans cette partie nous étudions la composition de deux relations topologiques ayant un "objet" en commun. Le but de notre recherche est de déduire l'ensemble des relations topologiques possibles entre deux objets connaissant les relations de ces deux objets avec un troisième objet. Nous avons réalisé cette recherche car elle nous semble être la base indispensable de tous travaux sur la gestion de la redondance et la vérification de la cohérence de l'information topologique d'un schéma CONGOO.

Notre étude formelle de la composition des relations est basée sur la théorie des ensembles car celle-ci nous paraît beaucoup plus simple à utiliser que la logique des prédicats.

Avant de commencer rappelons quelques définitions de la théorie des ensembles.

Egalité : $A = B \equiv \forall x\ (x \in A \equiv x \in B)$

Inclusion : $A \subseteq B \equiv \forall x\ (x \in A \Rightarrow y \in B)$

Union : $\forall x\ (x \in A \cup B \equiv x \in A \vee x \in B)$

Intersection : $\forall x\ (x \in A \cap B \equiv x \in A \wedge x \in B)$

Notre recherche sur la composition des relations topologiques est basée sur la règle de la transitivité de l'inclusion des ensembles et les règles connexes.

$A \subseteq B \wedge B \subseteq C \Rightarrow A \subseteq C$
$A \cap B \neq \emptyset \wedge B \subseteq C \Rightarrow A \cap C \neq \emptyset$

Pour pouvoir utiliser la théorie des ensembles nous donnons aux trois éléments géométriques une représentation par des ensembles munis de certaines caractéristiques.
Un point est représentée par un ensemble contenant un seul élément : $\forall P : \mathrm{Card}(P) = 1$
Une ligne est représenté par un ensemble composé de plusieurs éléments : $\forall L : \mathrm{Card}(L) > 1$
Le polygone est représenté par une union de deux ensembles de plusieurs éléments avec une intersection vide : $\forall S : S = I \cup F$ et $I \cap F = 0$ et $\mathrm{Card}(I) > 1$ et $\mathrm{Card}(F) > 1$

La démarche que nous avons élaborée pour trouver l'ensemble des relations possibles est en trois points. Tout d'abord, nous traduisons les deux relations topologiques connues et l'ensemble des relations définies entre les deux objets n'ayant pas de relation en commun par les concepts de la théorie des ensembles. Cette traduction est grandement facilitée par les définitions des relations topologiques vues ci-dessus (4.4). Ensuite, pour chaque relation de l'ensemble des relations définies nous vérifions, grâce à la règle de transitivité de l'inclusion des ensembles et des règles connexes, si elles peuvent exister ou non c'est-à-dire si elles peuvent faire partie de l'ensemble des relations possibles (qui tient compte des deux relations de départ). Toutes les relations qui peuvent exister sont donc ajoutées dans l'ensemble des relations possibles. Enfin, une fois que tout l'ensemble des relations définies est passé en revue nous retraduisons les concepts d'ensembles en relations topologiques.

Appliquons notre démarche de manière détaillée pour des relations entre points.
1. Seules les relations de St et de Sn sont définies entre les points. Remarquons que la relation de superposition totale est symétrique quand elle est appliquée entre points.
Quatre compositions sont donc possibles :
$St(x, y) \wedge St(y, z) \Rightarrow ?$
$Sn(x, y) \wedge St(y, z) \Rightarrow ?$
$St(x, y) \wedge Sn(y, z) \Rightarrow ?$
$Sn(x, y) \wedge Sn(y, z) \Rightarrow ?$
L'ensemble des relations définies entre deux points sont $Sn(x, z)$ et $St(x, z)$ et leurs symétriques.
Traduction : $St(x, y) \rightarrow P1 = P2$ et $Sn(x, y) \rightarrow P1 \cap P2 = \emptyset$
$(P1 = P2) \wedge (P2 = P3) \Rightarrow ?$
$(P1 \cap P2 = \emptyset) \wedge (P2 = P3) \Rightarrow ?$
$(P1 = P2) \wedge (P2 \cap P3 = \emptyset) \Rightarrow ?$
$(P1 \cap P2 = \emptyset) \wedge (P2 \cap P3 = \emptyset) \Rightarrow ?$

2. On remplace tour à tour une des relations de la formule par une de l'ensemble des relations définies et on vérifie s'il n'y a pas contradiction entre le résultat de cette composition et la relation remplacée par la règle de transitivité et les règles connexes. Si oui on ne l'insère pas dans l'ensemble des relations possibles.

Pour la première composition cela donne :

$(P1 = P3) \wedge (P2 = P3) \Rightarrow (P1 = P2)$ ok

$(P1 = P2) \wedge (P1 = P3) \Rightarrow (P2 = P3)$ ok

Nous remarquons que les deux formules sont justes, nous insérons donc la relation $(P1 = P3)$ dans l'ensemble des relations possibles.

$(P1 \cap P3 = \emptyset) \wedge (P2 = P3) \Rightarrow (P1 \cap P2 = \emptyset)$ or $(P1 = P2)$ absurde

La relation $(P1 \cap P3 = \emptyset)$ ne fait donc pas partie de l'ensemble des relations possibles. Chaque relation de l'ensemble des relations définies ayant été étudiée nous trouvons comme résultat

$(P1 = P3) \wedge (P2 = P3) \Rightarrow (P1 = P2)$

En utilisant le même procédé pour les autres compositions nous trouvons :

$(P1 \cap P2 = \emptyset) \wedge (P2 = P3) \Rightarrow (P1 \cap P3 = \emptyset)$

$(P1 = P2) \wedge (P2 \cap P3 = \emptyset) \Rightarrow (P1 \cap P3 = \emptyset)$

$(P1 \cap P2 = \emptyset) \wedge (P2 \cap P3 = \emptyset) \Rightarrow (P1 \cap P3 = \emptyset) \vee (P1 = P3)$

3. Nous retraduisons les compositions en termes de relations topologiques.

$St(x, y) \wedge St(y, z) \Rightarrow St(x, z)$

$Sn(x, y) \wedge St(y, z) \Rightarrow Sn(x, y)$

$St(x, y) \wedge Sn(y, z) \Rightarrow Sn(x, y)$

$Sn(x, y) \wedge Sn(y, z) \Rightarrow St(x, z) \vee Sn(x, z)$

Nous pouvons appliquer cette démarche pour les trois types d'éléments géométriques (point, ligne, polygone).

Appliquons à nouveau cette démarche entre trois objets représentés chacun par une ligne.

Les relations St, Sp et Sn sont définies entre deux lignes et seul la Sn est symétrique. Nous avons donc 25 compositions différentes et un ensemble de cinq relations définies.

Le résultat est :

$Sn(x, y) \wedge Sn(y, z) \Rightarrow Sn(x, z) \vee (Sp(x, z) \wedge Sp(z, x)) \vee (Sp(x, z) \wedge St(z, x)) \vee (St(x, z) \wedge Sp(z, x)) \vee (St(x, z) \wedge St(z, x))$

$Sn(x, y) \wedge Sp(y, z) \wedge Sp(z, y) \Rightarrow Sn(x, z) \vee (Sp(x, z) \wedge Sp(z, x)) \vee (St(x, z) \wedge Sp(z, x))$

$Sn(x, y) \wedge Sp(y, z) \wedge St(z, y) \Rightarrow Sn(x, z)$

$Sn(x, y) \wedge St(y, z) \wedge Sp(z, y) \Rightarrow Sn(x, z) \vee (Sp(x, z) \wedge Sp(z, x)) \vee (St(x, z) \wedge Sp(z, x))$

$Sn(x, y) \wedge St(y, z) \wedge St(z, y) \Rightarrow Sn(x, z)$

$Sp(x, y) \wedge Sp(y, x) \wedge Sn(y, z) \Rightarrow Sn(x, z) \vee (Sp(x, z) \wedge Sp(z, x)) \vee (Sp(x, z) \wedge St(z, x))$

$Sp(x, y) \wedge Sp(y, x) \wedge Sp(y, z) \wedge Sp(z, y) \Rightarrow Sn(x, z) \vee (Sp(x, z) \wedge Sp(z, x)) \vee (Sp(x, z) \wedge St(z, x)) \vee (St(x, z) \wedge Sp(z, x)) \vee (St(x, z) \wedge St(z, x))$

$Sp(x, y) \wedge Sp(y, x) \wedge Sp(y, z) \wedge St(z, y) \Rightarrow Sn(x, z) \vee (Sp(x, z) \wedge Sp(z, x)) \vee (Sp(x, z) \wedge St(z, x))$

$Sp(x, y) \wedge Sp(y, x) \wedge St(y, z) \wedge Sp(z, y) \Rightarrow Sn(x, z) \vee (Sp(x, z) \wedge Sp(z, x)) \vee (Sp(x, z) \wedge St(z, x)) \vee (St(x, z) \wedge Sp(z, x)) \vee (St(x, z) \wedge St(z, x))$

$Sp(x, y) \wedge Sp(y, x) \wedge St(y, z) \wedge St(z, y) \Rightarrow (Sp(x, z) \wedge Sp(z, x))$

$St(x, y) \wedge St(y, z) \wedge Sn(y, z) \Rightarrow Sn(x, z)$

$St(x, y) \wedge St(y, z) \wedge Sp(y, z) \wedge Sp(z, y) \Rightarrow (Sp(x, z) \wedge Sp(z, x))$

$St(x, y) \wedge St(y, z) \wedge Sp(y, z) \wedge St(z, y) \Rightarrow (Sp(x, z) \wedge St(z, x))$

$St(x, y) \wedge St(y, z) \wedge St(y, z) \wedge Sp(z, y) \Rightarrow (St(x, z) \wedge Sp(z, x))$

$St(x, y) \wedge St(y, z) \wedge St(y, z) \wedge St(z, y) \Rightarrow (St(x, z) \wedge St(z, x))$

$Sp(x, y) \wedge St(y, z) \wedge Sn(y, z) \Rightarrow (Sp(x, z) \wedge Sp(z, x))$

$Sp(x, y) \wedge St(y, z) \wedge Sp(y, z) \wedge Sp(z, y) \Rightarrow (Sp(x, z) \wedge Sp(z, x)) \vee (Sp(x, z) \wedge St(z, x)) \vee (St(x, z) \wedge Sp(z, x))$

$Sp(x, y) \wedge St(y, z) \wedge Sp(y, z) \wedge St(z, y) \Rightarrow (Sp(x, z) \wedge St(z, x))$

$Sp(x, y) \wedge St(y, z) \wedge St(y, z) \wedge Sp(z, y) \Rightarrow (Sp(x, z) \wedge Sp(z, x)) \vee (Sp(x, z) \wedge St(z, x)) \vee (St(x, z) \wedge Sp(z, x)) \vee (St(x, z) \wedge St(z, x))$

$Sp(x, y) \wedge St(y, z) \wedge St(y, z) \wedge St(z, y) \Rightarrow (Sp(x, z) \wedge St(z, x))$

$St(x, y) \wedge Sp(y, z) \wedge Sn(y, z) \Rightarrow Sn(x, z)$

$St(x, y) \wedge Sp(y, z) \wedge Sp(y, z) \wedge Sp(z, y) \Rightarrow Sn(x, z) \vee (Sp(x, z) \wedge St(z, x)) \vee (St(x, z) \wedge Sp(z, x))$

$St(x, y) \wedge Sp(y, z) \wedge Sp(y, z) \wedge St(z, y) \Rightarrow (Sp(x, z) \wedge St(z, x))$

$St(x, y) \wedge Sp(y, z) \wedge St(y, z) \wedge Sp(z, y) \Rightarrow (Sp(x, z) \wedge Sp(z, x)) \vee (Sp(x, z) \wedge St(z, x)) \vee (St(x, z) \wedge Sp(z, x)) \vee (St(x, z) \wedge St(z, x))$

$St(x, y) \wedge Sp(y, z) \wedge St(y, z) \wedge St(z, y) \Rightarrow (St(x, z) \wedge Sp(z, x))$

Nous n'avons fait qu'une partie du travail mais il suffit d'appliquer notre démarche entre toutes les combinaisons des éléments géométriques pour déduire le résultat de toutes les compositions possibles de deux relations.

4.7. Conclusion

Dans ce chapitre, nous avons constaté qu'il restait beaucoup de recherches à effectuer sur la formalisation des relations topologiques et surtout sur l'ensemble des relations topologiques qui doivent être distinguées.

A propos du choix des relations topologiques utilisées dans le modèle CONGOO, nous avons souligné que son principal avantage est qu'il repose sur deux relations majeures seulement relativement intuitives, ce qui facilite leurs utilisations.

Pour terminer, soulignons que notre définition formelle des relations topologiques de voisinages et de superpositions, nous a permis de découvrir les propriétés de ces relations et de réaliser leur composition.

5. Conclusion

Dans le but de faciliter le développement des systèmes d'information géographique, une méthode de conception, appelée MECOSIG a été créé par Pantazis. La création de cette méthode comprend notamment l'élaboration d'un nouveau modèle de structuration de l'information géographique appelée CONGOO.

Pour étudier en détail ce nouveau modèle, nous avons tout d'abord essayé de cerner dans le premier chapitre les spécificités des systèmes d'information géographique par rapport à un système d'information "traditionnel". Nous avons relevé que sa grande spécificité est de traiter conjointement des données ayant une représentation graphique référencée spatialement (appelées données géo-graphiques) et des données alphanumériques (référencées spatialement ou pas). Ensuite nous avons montré que l'informatisation des systèmes d'infomation géographique demande des solutions informatiques différentes des systèmes d'information "traditionnels" vu les caractéristiques particulières des données qu'il traîte.

Dans le chapitre deux, après avoir rappelé les objectifs d'un modèle conceptuel de structuration de l'information, nous avons identifié les besoins d'un modèle conceptuel de structuration de l'information géographique en présentant les lacunes du modèle E-A (Entité - Association) pour cette information. Nous avons pu ainsi mettre en évidence qu'un modèle de structuration de l'information géographique devait notamment posséder une grande expressivité et des mécanismes de structuration puissants. Après, nous avons observé que le modèle CONGOO répondait bien aux besoins identifiés mais nous avons remarqué qu'il ne possède pas de sémantique formelle.

Nous avons donc commencé à lui donner une sémantique formelle en utilisant un langage formel de spécification appelé GLIDER. Ce travail a permis de relever quelques ambiguités et imprécisions dans les définitions des concepts du modèle CONGOO. Ceci souligne l'importance d'avoir une sémantique formelle du modèle. De plus, la définition formelle des relations topologiques a permis d'étudier plus en détail ces relations et d'en déduire formellement leurs propriétés ainsi que leurs compositions. Ceci nous semble très important car ces propriétés et ces compositions vont permettre d'effectuer des recherches sur la vérification et le maintien de la cohérence ainsi que la gestion de la redondance de l'information topologique des schémas CONGOO.

Nous terminons cette conclusion en soulignant qu'il reste de nombreuses recherches à effectuer dans le domaine des systèmes d'information géographique. Nous pensons à ce propos que les travaux de recherche dans ce domaine ont plusieurs années de retard sur les travaux effectués dans le domaine des systèmes d'informations de l'informatique de gestion.

Pour rattraper ce retard soulignons qu'il ne suffit pas d'appliquer aveuglément les solutions des systèmes d'information traditionnels mais qu'il faut les adapter en connaissant les spécificités des systèmes d'information géographique et de l'information qu'ils traitent. Notre travail a notamment montré que des recherches approfondies devaient être réalisées sur les méthodes de conception des systèmes d'information géographique et surtout sur la conception d'ateliers de génie logiciel pour assister ces méthodes. Des travaux importants devraient être effectués aussi sur les relations topologiques et plus particulièrement sur la gestion de la redondance et sur la vérification de la cohérence de l'information topologique.

Bibliographie

Bibliographie citée

[Abiteboul et Hull 87] Abiteboul S., Hull R.,"IFO : A formal semantic database model", ACM Transactions on Database Systems, Vol. 12, N°4, pp. 525-565, 1987.

[Abler 87] Abler R. "The National Science Foundation National Center for Geographic Information and Analysis", Int. J. Geographical Information Systems, Vol. 1, pp. 303-326, 1987.

[Allen 83] Allen J., "Maintaining Knowledge about Temporal Intervals", Communications of the ACM, Vol. 26, N°11, pp. 832-843, November 1983.

[Aronoff 89] Aronoff S., "Geographic information systems : a management perspective", WDL Publications, Ottawa, Canada, 1989.

[Barr 87] Barr S., "Expériences de topologie", Traduction française, de la lysimaque, Paris, 1987.

[Bédart et Caron 93] Bédart Y., Caron C., "Bases de données à référence spatiale (BDRS)", extrait de cours postgrade en informatique sur les nouvelles approches pour les bases de données, Ecole Polytechnique Fédérale de Lausanne, 1993.

[Bodart 93] Bodart F., "Conception des systèmes d'information", Notes de cours, Institut d'informatique, Namur, 1993.

[Bodart et Pigneur 93] Bodart F., Pigneur Y., "Conception assistée des systèmes d'information, méthode - modèles - outils", 2 éd. Masson, 1993.

[Calloz 90] Calloz R.,"Système d'information géographique", 1 et 2, Notes provisoires du cours du 3ème cycle "Environnement", Ecole Polytechnique Fédérale de Lausanne, Suisse, 1990.

[Caron, Bédart et Gagnon 1993] Caron C., Bédart Y., Gagnon P., "MODUL-R : un formalisme individuel adapté pour les SIRS", Revue de géomatique, Vol. 3, N°3, pp. 283-306, 1993.

[Chang, Jeugert et Li 89] Chang S., Jeugert E., Li Y., "The design of pictorial databases based upon the theory of symbolic projections", Proc. of Symposium on the Design and Implementation of large Spatial Databases, Lecture Notes in Computer Science, Springer-Verlag, Santa Barbara(USA), N°409, pp. 303-323, 1989.

[Chen 76] Chen P.P., "The Entity-Relationship Model : Toaward a Unified View of Data, ACM TODS, Vol. 1, N°1, 1976.

[Clarke 81] Clarke B," A Calculus of Individuals Based on 'Connection' ", Notre Dame Journal of Formal Logic, Vol. 22, N°3, pp. 204-218, july 1981.

[Clarke 85] Clarke B," Individuals and Points", Notre Dame Journal of Formal Logic, Vol. 26, N°1, pp. 61-75, january 1985.

[Clementini, Di Felice et Van Oosterom 93] Clementini E., Di Felice P., Van Oosterom P.," A Small Set of Formal Topological Relationships Suitable for End-User Interaction", Proc. of the Third Int. Symposium on Advances in Spatial Databases, SSD'93, Lecture Note in Computer Science, Springer-Verlag, Singapore, N°692, pp. 277-295, june 1993.

[Coad et Yourdon 93] Coad P., Yourdon E., "Analyse orientée objet", Traduction française de Abdelkader Boughlam, Masson, Paris, Prentice Hall, London, 1993.

[Cohn et Gotts 94] Cohn G., Gotts N.," A Theory of Spatial Regions With Indeterminate Boundaries", papers from the Workshop 'Topological Foundations of Cognitive Science' at the First Int. Summer Institute in Cognitive Science, Buffalo, july 1994.

[Cui, Cohn et Randell 93] Cui Z., Cohn A.G., Randell D.A., "Qualitative and Topological Relationships in Spatial Databases", Proc. of the Third Int. Symposium on Advances in Spatial Databases, SSD'93, Lecture Notes in Computer Science, Springer-Verlag, Singapore, N°692, pp. 296-315, june 1993.

[David et al 93] David B., Raynal L., Schorter G., Mansart V., "Géo2 : modélisation objet de données géographiques", Revue de géomatique, Vol. 3, N°3, pp. 265-281, 1993.

[de Hoop et van Oosterom 92] de Hoop S., Van Oosterom P., "Storage and manipulation of topology in Postgres", Proc. of Third European Conference on Geographical Information Systems, EGIS'92, Munich(Germany), pp. 1324-1336, 1992.

[de Hoop, van Oosterom et Molenaar 93] de Hoop S., Van Oosterom P., Molenaar M., "Topological Querying of Multiple Map Layers", Proc. of the European Conference on Spatial Information Theory, COSIT'93, Lecture Notes in Computer Science, Springer-Verlag, Italy, N°716, pp. 139-157, September 1993.

[Donnay 92] Donnay J.-P., "Les systèmes d'information géographique", Communication présentée au séminaire "Les systèmes d'information géographique dans l'administration publique", organisé par le Ministère de l'Aménagement du Territoire du Grand-Duché de Luxembourg et l'Institut Européen d'Administration Publique (IEAP), Luxembourg, 11-12 Novembre, 1992.

[Dubois, Hagelstein et Rifaut 90] Dubois E. , Hagelstein J. , Rifaut A., "ERAE : a Formal Language for Expressing and Structuring Real-Time Requirements", Philips Research Labotory, Manuscript 353, Louvain-la -Neuve, June 1990.

[Dubois, Du Bois, Rifaut et Wodon 91] Dubois E. , Du Bois P. , Rifaut A., Wodon P. , "Glider User Manual", Esprit Project ICARUS 2537, Task-spec-Func-028-R, Namur-Brussel, July 1991.

[Dubois, Du Bois et Petit 93] Dubois E., Du Bois P., Petit M., "O - O Requirements Analysis : an Agent Perspective", Proc. of the Seventh European Conference on Object-Oriented Programming, ECOOP'93, Lecture Notes in Computer Science, Springer-Verlag, Kerserlautern(Allemagne), pp. 458-479, July 1993.

[Egenhofer 89] Egenhofer M., "A formal Definition of binary topological Reltionships", Proc. Third Int. Conference on Foundations of Data Organization and Algorithms, Lecture Notes in Computer Science, Springer-Verlag, Paris(France), N°367, pp. 457-472, June 1989.

[Egenhofer, Franck et Jackson 89] Egenhofer M., Franck A., Jackson J., "A Topological Data Model for Spatial Databases ", Proc. of First Symposium on the Design and Implementation of large Spatial Databases, SSD'89, Lecture Notes in Computer Science, Springer-Verlag, Santa Barbara(USA), N°409, pp. 271-286, 1989.

[Egenhofer 91] Egenhofer M., "Reasoning about Binary Topological Relations", Proc. of the 2nd Sympsium on Advances in Spatial Databases, SSD'91, Lecture Notes in Computer Science, Springer-Verlag, Zurich(Switzerland), N°525, pp. 143-160 august, 1991.

[Egenhofer et Franzosa 91] Egenhofer M, Franzosa R. D., "Point-set topological spatial relations", Int. J. of Geographical Information Systems, Vol. 5, N°2, pp. 161-174, 1991.

[Egenhofer et Franzosa 95] Egenhofer M., Franzosa R., "On the equivalence of topological relations", Int. J. of Geographical Information Systems, Vol. 9, N°2, pp. 133-152, 1995.

[Egenhofer et Al-taha 92] Egenhofer M., Al-Taha K., "Reasoning about Gradual Changes of Topological Relationship", Proc. of the Int. Conference GIS on Theories and Methods of Spatio-Temporal Reasoning in Geographic Space, Lecture Notes in Computer Science, Springer-Verlag, Pisa(Italy), N°639, pp. 196-219, September, 1992.

[Egenhofer et Mark 95] Egenhofer M., Mark D., "Modelling Conceptuel neighbourhoods of topological line- region relations", Int. J. of Geographical Information Systems, Vol. 9, N°5, pp. 555-565, 1995.

[Egenhofer et Sharma 93] Egenhofer M., Sharma J., "Topological Relations Between Regions in R2 and Z2", Proc. of 3th Int. Symposium on Advances in Spatial Databases, SSD'93, Lecture Notes in Computer Science, Springer-Verlag, Singapore, N°692 , pp. 316-336, June 1993.

[Egenhofer et Herring 95] Egenhofer M., Herring J., Proc. of 4th Int. Symposium on Advances in Spatial Databases, SSD'95, Lecture Notes in Computer Science, Springer-Verlag, Portland(USA), N°951 , pp. 1-165, August 1995.

[Egenhofer, Clementini et Di Felice 94] Egenhofer M., Clementni E., Di Felice P.,"Topological relations between regions with holes", Int. J. Geographical Information Systems, Vol. 18, N°2, pp. 129-142, 1994.

[Hadzilacos et Tryfona 91] Hadzilacos T., Tryfona N., "A Conceptual Schema for Geographic Databases", Proc. of Second European Conference Geographical Information Systems, EGIS'91, Brussels(Belgium), pp. 397-405, April 1991.

[Hadzilacos et Tryfona 92] Hadzilacos T., Tryfona N., "A Model for Expressing Topological Integrity Constraints in Geographic Databases", Proc. of the Int. Conference GIS on Theories and Methods of Spatio-Temporal Reasoning in Geographic Space, Lecture Notes in Computer Science, Springer-Verlag, Pisa(Italy), N°639, pp. 252-268, September 1992.

[Hainaut 94] Hainaut J.L., "Conception base de données matière approfondie", Notes de cours, Institut d'Informatique, Namur, 1994.

[Herring 91] Herring J., "The mathematical modeling of spatial and non-spatial information in geographic information systems", In : Cognitive and Linguistc Aspects of Geographic Space, éd. par Mark D. et Franck A., 1991.

[Goodchild 91] Goodchild M., "Integrating GIS and Environmental Modeling at Global Scales", Proc. GIS/LIS'91, Vol.1, pp.117-127, 1991.

[Greenspan, Bordiga et Mylopoulos 86] Greenspan S., Bordiga A., Mylopoulos J., "A requirements modeling language and its logic". In Bodie M. L. et Mylopoulos J., editors, On knowledge base management systems, Topics in information systems, pp. 471-502, 1986.

[Guenther 90] Guenther O., Buchman A., "Research Issue in Spatial Databases", SIGMOD RECORD, Vol. 19, N°4, pp. 61-68, December 1990.

[Güting 88] Güting R., "Géo-Relational Algebra: A Model and Query Language for Geometric Database Systems", Proc. of Int. Workshop on Computational Geometry and its Applications' CG'88, Lecture Notes in Computer Science, Springer-Verlag, N°333, pp. 90-97, March 1988.

[ICARUS 90] "First Version of the ICARUS Product and Process Language for the Specification of the System Functionalities", March 1990.

[Jaton et Bédard 93] Jaton A., Bédard Y., "La géomatique, une industrie en restructuration : l'exemple du Québec et du canada", Compte rendu des actes du Séminaire SIT-93, 26 Mars, Lausanne, pp. 63-74, 1993.

[Langlois 94] Langlois P.," Formalisation des concepts topologiques en géomatique", Revue de géomatique, Vol. 4, N°2, pp. 181-205, 1994.

[Larue, Pastre et Viémont 93] Larue T., Pastre D., Viémont Y., "Strong Integration of Spatial Domains and Operators in a Relational Database System", Proc. of Third Int. Symposium on Advances in Spatial Databases, SSD'93, Lecture Notes in Computer Science, Springer-Verlag, Singapore, N°692, pp.53-72, June 1993.

[Laurini et Milleret-Raffort 93] Laurini R., Milleret-Raffort F.," Les bases de données en géomatique", Hermes, Paris, 1993.

[Marx 91] Marx R.,"TIGER and GIS : Linking People and Institutions to your Map Base",1991-92 Int. GIS Sourcebook, Forts Collins, CO : GIS World, Inc., pp.413-416, 1991.

[Math 80] "Petite encyclopédie des mathématiques ", Traduction française de Lions J.-L., éd. K. Pagoulatos, Paris, 1980.

[Molenaar, Kufoniyi et Boulocos 94] Molenaar M., Kufoniyi O., Bouloucos T., "Modelling topologic relationships in vector maps", Proc. of Sixth Int. Symposium on Spatial Data Handling, Vol. 1, pp. 112-125, 1994.

[Pageau, Bédard et Gagnon 93] Notes de cours sur MODUL-R version 2.0. Centre de recherche en géomatique, Université Laval, 1993.

[Pantazis 94a] Pantazis D., "La méthode de conception de S.I.G. ME.CO.S.I.G. et le formalisme ME CON.G.O.O.", Proc. of fifth European conference and exhibition on Geographical Information Systems ,EGIS/MARI'94, Paris(France), Avril 1994.

[Pantazis 94] Pantazis D., "Analyse méthodologique des phases de conception et de developpement d'un système d'information géographique", Thèse de doctorat, Liège, 1994.

[Pantazis et Donnay] Pantazis D., Donnay J.-P.,"MECOSIG et CONGOO, méthode et formalisme de conception de S.I.G.", Hermes, en préparation.

[Pornon 93] Pornon H., "Quelques réflexions sur la difficulté d'utiliser MERISE pour la modélisation des bases de données géographiques", Revue de géomatique, Vol. 3, N°3, pp. 245-255, 1993.

[Pullar 88] Pullar D. "Data definition and operators on a spatial data model", Proc. of ACSM-ASPRS Annual Convention, held in St Louis, Missouri, pp.197-202, 1988.

[Randell, Cui et Cohn 92] Randell D., Cui Z., Cohn A., "A Spatial Logic based on Regions and Connection", Proc. 3rd Int. Conf on Knowledge Representation and Reasoning, Boston, October, 1992.

[Rouet 91] Rouet M., "Les données dans les systèmes d'information géographique", Hermes, Paris, 1991.

[Smith 87] Smith T., Peuquet D., Menon S., Agrawal P.,"KBGIS-2 : a knowledge-based geographical information system", Int. J. Geographical information Systems, Vol. 1, 149-172, 1987.

[Smith, Park 92] Smith T., Park K.," Algebraic approach to Spatial reasoning", Int. J.Geographical Information Systems, Vol. 6, N°3, pp. 177-192, 1992.

[Topo 81] "Topologie, cours et problèmes", Traduction française, Mcgraw-Hill, Paris, 1981.

[Vieu 93] VIEU L., "A Logical Frame work for Reasoning about Space", Proc. of the European Conference on Spatial Information Theory, COSIT'93, Lecture Notes in Computer Science, Springer-Verlag, Italy, N°716, pp. 25-35, Septembre 1993.

[Wagner 88] Wagner D.,"A method of evaluating polygon overlay algorithms", Proc. of ACSM-ASPRS Annual Convention, held in St Louis, Missouri, pp.173-183, 1988.

[Worboys et al 90] Worboys M., Hearnshaw H., Maguire D., "Object-oriented data modelling for spatial databases", Int. J. Geographical Information Systems, Vol. 4, N°4, pp. 369-383, 1990.

Bibliographie consultée

Collet C., "Systèmes d'information géographiques en mode image", Presses polytechniques et universitaires romandes, Lausanne, Suisse, 1992.

Laurini R., Thompson D., "Fundamentals of spatial infomation systems", The Apic series, Academic Press, Great Britain, 1993.

Medeiros C., Pires F., "Databases for GIS", SIGMOD RECORD, Vol. 23, N°1, pp.107-114, March 1994.

Meyer B., "On Formalism in Specification", IEEE SOFTWARE, pp. 6-26, January 1985.

Pantazis D., "Conception d'un système d'information à référence spatiale pour la surveillance et le contrôle de la qualité des eaux fluviales", Rapport de recherche, Institut d'Aménagement des Terres et des Eaux - Département de génie rural, Ecole Polytechnique de Lausanne, Suisse, 1991.

Pantazis D., "Conception d'un système d'information servant au contrôle de la qualité des eaux des rivières d'un canton et destiné à un laboratoire cantonal", Travail pour l'obtention du certificat d'études du 3éme cycle : La gestion des systèmes d'information : outils et méthodes, Institut des Hautes Etudes en Administration Publique, Lausanne, 1991.

Tanzi T., Ubeda T., "Contrôle topologique de la cohérence dans les bases de données géographiques : application aux réseaux", Revue de géomatique, Vol. 5, N°2, pp. 131-155, 1995.

RELATIONS

- LOGIQUES
  notation graph.(+ cardinalités)
  appliquées entre objets, classes, couches, sous-couches et leurs combinaisons (avec application ou non des contraintes)
- DE STRUCTURE (structure topologique et / ou sémantique) (avec application ou non des contraintes)
  - Gén.-Spéc.
    appliquées ente classes d'objets seulement
    notation graph.
  - Composé - Composants
    appliquée entre instances, objets, classes, couches, sous-couches et leurs combinaisons
    notation graph.
    (+ cardinalités, si appliquée entre objets)
    - *Contenant-Contenu*
    - *Collection-membre*
    - *Collection-membre avec contraintes*
- TOPOLOGIQUES appliquées entre instances, objets, classes, couches, sous-couches, et leurs combinaisons (avec application ou non des contraintes)
  - Permises — (+) S(p,t,n) V(p,t,n)
    (transformées en contraintes d'intégrité)
    (peut exister). Notation graph. (*) : coin inférieur gauche des objets (elles concernent la même instance). Coin inférieur gauche des classes, couches, sous-couches (elles concernent toutes les instances représentées par l'objet ou les objets inclus dans la classe, couche, sous-couche).
  - Obligatoires — (+) S(p,t,n) V(p,t,n) ; (-) S(p,t,n) V(p,t,n)
    (doit exister)
    notation graph. : a) idem (*), mais dans le coin inférieur droit
    b) —[relation]—
    (+ cardinalités, si appliquées entre objets) ou coin inférieur droit des classes, couches, sous-couches

Notes : - Les relations topologiques obligatoires peuvent être orientées et appliquées de manière combinatoire en utilisant les conjonctions de coordination «et» et «ou». - Les relations topologiques obligatoires peuvent être appliquées simultanément à la relation composé - composant entre différents éléments. Elles seront notées avec un carré au-dessous du triangle.

Les "relations" proposées dans le cadre du formalisme CONGOO [Pantazis et Donnay, p.69]

Lecture sommaire du schéma conceptuel de données de la figure 2.11 réalisée par Pantazis [Pantazis et Donnay; annexe 1, p. 6]

Le schéma contient quatre classes d'objets géo-graphiques. "La classe ETAT ne contient qu'une seule instance (OGC). Les classes COMMUNES et REGIONS D'AMENAGEMENT contiennent des OGC-Pol". La classe SECTIONS CADASTRALES contient des OGS-Pol.

"Entre les instances de la classe 'COMMUNES', les relations topologiques de voisinage partiel et nul sont permises : +Vp,n. Sont interdites les relations de superposition partielle et totale, et de voisinage total : -Sp,t, -Vt. Idem pour les instances des classes SECTIONS CADASTRALES, REGIONS D'AMENAGEMENT".

"La classe REGIONS D'AMENAGEMENT est superposée à la classe d'une seule instance ETAT et vice versa : St. Cela signifie que l'on a aussi Vt entre les deux classes; l'Etat est composé par l'ensemble des instances appartenant à la classe REGIONS D'AMENAGEMENT (application simultanée de la relation composé-composant et de la relation topologique obligatoire St)".

"Chaque région d'aménagement est superposée partiellement par chacune des communes qui la compose (1,N); chaque commune est superposée totalement à la région d'aménagement, à laquelle elle participe en tant que composant."

"Chaque région d'aménagement est superposée partiellement par chacune des sections cadastrales qui les composent (1,N); Chaque section cadastrale est superposée totalement à la région d'aménagement, à laquelle elle participe en tant que composant."

"Chaque commune est superposée partiellement par chacunes des sections cadastrales qui les composent (1,N); chaque section cadastrale est superposée totalement à la commune à laquelle elle participe en tant que composant."

"La classe SECTIONS CADASTRALES est superposée totalement à la classe COMMUNES : St."

"Les classes SECTIONS CADASTRALES et COMMUNES sont superposées totalement à l'instance ETAT; de plus, chacune d'entre elles la compose."

"Des lignes d'exclusivité déterminent le nombre de composants attribués aux objets composés Etat, Commune, Région d'Aménagement."